

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Vizualizace molekul na mobilním zařízení
Visualization of Molecules on Mobile Devices

2012

Bc. Zdeněk Vjater

Zadání diplomové práce

Student: **Bc. Zdeněk Vjater**
Studijní program: **N2647 Informační a komunikační technologie**
Studijní obor: **2612T059 Mobilní technologie**
Téma: **Vizualizace molekul na mobilním zařízení**
Visualization of Molecules on Mobile Devices

Zásady pro vypracování:

Cílem práce je napsat aplikaci, která bude číst popisy molekul z několika známých formátů souborů s popisem molekul a vizualizovat je ve 2D a 3D zobrazení na mobilním zařízení.

1. Zmapujte možnosti 3D vizualizace na různých mobilních platformách.
2. Zjistěte, jaké existující aplikace pro 3D vizualizaci molekul existují na platformě PC a na mobilních platformách.
3. Vyberte vhodnou platformu pro implementaci (Java ME, Windows CE, Android, ...) a výběr zdůvodněte.
4. Proveďte analýzu a návrh výsledné aplikace a případné knihovny. Aplikace bude schopna importovat a vizualizovat alespoň 3 různé běžně používané formáty pro uložení molekul, definujte i API pro přidání importu dalšího formátu.
5. Aplikaci implementujte, otestujte a srovnajte s reprezentativním vzorkem existujících aplikací.

Seznam doporučené odborné literatury:

1. Fox D., Box J.: Building solutions with the Microsoft .NET Compact Framework: architecture and best practices for mobile development, Addison-Wesley Publishing. ISBN: 0321197887.
2. Meier, R. Professional Android 2 Application Development. Wrox Press Ltd., 2010. ISBN: 978-0-47056-552-0.
3. Wells, M. J.: J2ME Game Programming (Game Development), Course Technology PTR, 2004, ISBN: 978-1-59200-118-7.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Pavel Moravec, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012

prof. RNDr. Vladimír Vašínek, CSc.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne 2.5.2012:

Handwritten signature of Jan Bárta in cursive script.

Poděkování

Rád bych poděkoval Ing. Pavlovi Moravcovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této diplomové práce.

Abstrakt

Tato práce se zabývá možnostmi vizualizace molekul na základě jejich geometrie, jež je popsána v souborech různých formátů. Zobrazování molekul má probíhat jak ve 2D tak 3D prostoru. V práci jsou popsány obecně používané zobrazovací modely a také některé známé programy řešící tuto problematiku. Vizualizace bude prováděna na mobilním zařízení a v práci je zdůvodněn výběr konkrétní použité platformy. Výsledná aplikace je schopná importu tří různých formátů a následné vizualizace molekuly, jež je v nich popsána. Tato aplikace je úspěšně otestována na reálném zařízení a vykreslovacími vlastnostmi je srovnatelná s již existujícími aplikacemi.

Klíčová slova

Android, molekula, atom, vazba, PDB, CML, XYZ, MDL, 2D vizualizace, 3D vizualizace, Jmol

Abstract

This project goes into possibilities of molecule visualisation based on their own geometry , which is described in files of various formats. Displaying of these molecules is available in 2D and also in 3D field. On the one hand, this work presents generally used displaying models, on the other hand, there can be also found some more known programs which solve this issue. Visualization will proceed in the mobile device and this work describe chosen platform. Final application is able to import three different formats and afterwards visualization of molecules, which are described in it. This application was successfully tested on the real device and with its drawing properties, it is comparable with already existing applications.

Key words

Android, molecule, atom, bond, PDB, CML, XYZ, MDL, 2D visualization, 3D visualization, Jmol

Seznam použitých symbolů, zkratk a termínů

CML	Chemical Markup Language
JavaME	Java Micro Edition
NIH/NCI	National Institutes of Health National Cancer Institute
PDB	Protein Data Bank
XML	Extensible Markup Language

Obsah

1 Úvod.....	1
2 Formáty pro popis geometrie molekul.....	2
2.1 PDB – Protein Data Bank format.....	2
2.2 CML – Chemical Markup Language.....	2
2.3 MDL Molfile.....	3
2.4 XYZ formát.....	4
2.5 Další formáty	5
2.6 Vzájemný převod formátů.....	5
3 Grafické znázornění molekul.....	7
3.1 3D modely molekul.....	7
3.1.1 Drátěný model.....	7
3.1.2 Tyčinkový model.....	8
3.1.3 Kuličkový model.....	8
3.1.4 Kalotový model.....	9
3.2 2D modely znázorňování molekul.....	10
3.2.1 Rozvinutý konstituční vzorec.....	10
3.2.2 Racionální konstituční vzorec.....	10
3.2.3 Geometrický vzorec.....	11
4 Aplikace pro 3D vizualizaci molekul.....	12
4.1 Rasmol.....	12
4.2 Molegro Molecular Viewer.....	12
4.3 Jmol.....	12
5 Možnosti 3D vizualizace na mobilních platformách.....	14
5.1 JavaME.....	14
5.2 Windows Mobile, Windows Phone.....	14
5.3 Android.....	15
6 Analýza programu.....	16
6.1 Výběr platformy.....	16
6.2 Základní funkce programu.....	16
6.3 Import souborů.....	17
6.4 2D vizualizace.....	18
6.5 3D vizualizace.....	19
6.6 Přepínání režimů vizualizace.....	22
7 Implementace.....	23
7.1 Základní struktura aplikace.....	23
7.2 Souborový prohlížeč.....	23

7.3	Atomy a vazby.....	24
7.3.1	Třída Atom.....	24
7.3.2	Třída Bond.....	25
7.4	Importní třídy.....	25
7.4.1	Společné rozhraní.....	25
7.4.2	Formát XYZ.....	26
7.4.3	Formát PDB.....	26
7.4.4	Formát CML.....	27
7.5	Příprava vizualizace.....	28
7.6	2D vizualizace.....	28
7.7	3D vizualizace.....	30
8	Testování aplikace.....	32
8.1	Testování v emulátoru.....	32
8.2	Testování na reálném zařízení.....	34
8.3	Srovnání testů.....	37
8.4	Porovnání aplikace s Jmol.....	37
9	Závěr.....	40
10	Použitá literatura.....	41
11	Přílohy.....	42

1 Úvod

Dnešní mobilní zařízení jsou již schopná pokročilé práce s 2D a 3D grafikou. Mezi obtížnější úlohy v této oblasti patří vizualizace molekul o různém počtu atomů. Kromě samotného vykreslování je u těchto zařízení možno využít dalších vlastností, jako je dotykové ovládání či detekce pohybu. Některé aplikace zabývající se vizualizací molekul jsou již dostupné i pro mobilní zařízení, ale většina z nich podporuje pouze 3D vizualizaci nebo se zabývá 2D vyobrazením chemických vzorců. V průběhu této práce bude vytvořena aplikace, jež by zahrnovala jak 2D tak i 3D vizualizaci, přičemž molekuly budou zobrazovány na základě geometrie popsané v jejich vstupních souborech.

První kapitola této práce se zabývá samotnými vstupními soubory, jež molekuly popisují. Ty mohou být zapsány v různých formátech, které se od sebe navzájem odlišují nejen stylem zápisu, ale i tím, jaké prvky molekuly popisují. Jednotlivé formáty lze mezi sebou navzájem převádět.

Další část práce popisuje používané zobrazovací modely pro 3D a 2D vizualizaci. Zatímco 3D modely většinou vycházejí z reálné podoby molekuly, 2D zobrazení bývá často určeno pouze chemickým vzorcem molekuly.

Čtvrtá kapitola obsahuje informace o několika základních programech využívajících 3D zobrazení pro vizualizaci molekul. Zmiňuje také, pro jakou platformu jsou tyto aplikace dostupné.

Dále jsou probrány možnosti 3D vizualizace na jednotlivých mobilních platformách. Jejich výčet je omezen na ty nejpoužívanější.

Následující kapitola se zabývá analýzou aplikace, jež má být v rámci práce vytvořena. Volí se zde implementační platforma a probírají se vlastnosti, které by implementovaná aplikace měla mít.

Sedmá kapitola již popisuje samotnou implementaci programu. Zahrnuje výčet základních stavebních prvků a jejich strukturu. Zabývá se nejen částí starající se o import vstupních souborů, ale také těmi, jež se následně starají o vizualizaci ve 2D i 3D prostoru.

Poslední část hodnotí vytvořený program. Posuzuje jeho funkčnost a také rychlost zpracovávání vstupních dat i vykreslování molekul. Probíhá zde i porovnání s aplikací Jmol, jež je dostupná pro stejnou platformu. Srovnávanými parametry jsou jejich výstupy i rychlost.

2 Formáty pro popis geometrie molekul

2.1 PDB – Protein Data Bank format

Tento formát je nejvíce využíván pro popis proteinů projektem Protein Data Bank. Soubor obsahuje kromě samotného popisu struktury molekuly další informace zapsané v jeho hlavičce, které jsou při vykreslování ignorovány. Bývají to například data o autorovi, či programu, jímž byla molekula vytvořena. Jsou označovány jako poznámky (REMARK).

Atomy jednotlivých prvků jsou zde uvozovány značkou ATOM. Následuje pořadové číslo, chemická značka daného prvku a jeho pozice v kartézském souřadnicovém systému. Pro kompletní popis molekuly je zapotřebí definovat také vazby mezi jednotlivými atomy. Ty jsou popisovány až po výčtu všech atomů. Také mají svou uvozovací značku, konkrétně slovo CONECT. Výčet vazeb se pojí k jednotlivým atomům. Není tedy definována každá vazba samostatně, ale jako první je zapsáno číslo atomu, ze kterého dané vazby vychází a až poté následují čísla všech atomů, do kterých tyto vazby vedou. Zapisování vícenásobných vazeb je dáno zápisem této vazby z obou stran. Vazba je tedy definována z prvního atomu do druhého i následně z druhého do prvního. Více o struktuře tohoto formátu se lze dočíst zde [1].

```

HEADER      PROTEIN
COMPND      WATER
ATOM        1   O   UNK      1      0.000      0.000      0.000      1.00      0.00
ATOM        2   H   UNK      1     -0.760     -0.584      0.000      1.00      0.00
ATOM        3   H   UNK      1      0.760     -0.584      0.000      1.00      0.00
CONECT      1      2      3
CONECT      2      1
CONECT      3      1
MASTER      0      0      0      0      0      0      0      0      3      0      3
END

```

Obrázek 2.1: Molekula vody zapsaná ve formátu PDB

2.2 CML – Chemical Markup Language

Jedná se o strukturu podobnou jazyku XML. Stejně jako v něm, i zde jsou definovány jednotlivé elementy, ve kterých mohou být vnořeny další. Existují párové a nepárové elementy, z nichž jedny vyžadují ukončující element, zatímco druhé stojí samostatně. Celá jedna molekula začíná párovým znakem <molecule>, jenž může obsahovat i identifikaci molekuly, tedy většinou její název nebo chemickou značku. Dále je soubor rozdělen na seznam jednotlivých

atomů (<atomArray>) a na seznam vazeb (<bondArray>). Ty již obsahují jednotlivé prvky, jimiž jsou atomy a vazby.

Každému atomu je přiřazena zvolená identifikace, jež se užívá při definování vazeb k určení, které dva atomy má každá z nich spojit. U tohoto formátu se definuje každá vazba zvlášť a ne všechny vazby pro jeden atom zároveň, jako tomu bylo u PDB. Definice atomů může obsahovat kromě jejich pozice, identifikace a označení prvku i další definované informace, jako je například izotopové číslo. Rozlišování vazeb mezi nimi je taktéž dáno parametrem, jenž se udává při definici vazby. Na jeho základě se určuje, jestli se jedná o jednoduchou vazbu nebo vícenásobnou. Podrobné informace o tomto formátu lze získat zde [2].

```
<molecule>
  <atomArray>
    <atom elementType="O" x3="-2.0" y3="0.0" z3="-1.0" id="a1">
    </atom>
    <atom elementType="H" x3="0.0" y3="5.0" z3="0.0" id="a2">
    </atom>
    <atom elementType="O" x3="2.0" y3="0.0" z3="1.0" id="a3">
    </atom>
  </atomArray>
  <bondArray>
    <bond id="b1">
      <string builtin="atomRef">a1</string>
      <string builtin="atomRef">a2</string>
      <string builtin="order">1</string>
    </bond>
    <bond id="b2">
      <string builtin="atomRef">a2</string>
      <string builtin="atomRef">a3</string>
      <string builtin="order">1</string>
    </bond>
  </bondArray>
</molecule>
```

Obrázek 2.2: Molekula vody zapsaná ve formátu CML

2.3 MDL Molfile

Tento formát má dvě volitelné přípony mdl a mol. V případě využití kterékoli z těchto přípon zůstává formát výsledného souboru stejný. Svými parametry a systémem popisu jednotlivých atomů a také vazeb se velice podobá formátu PDB. Zde se však nepoužívají žádné uvozovací znaky. O tom, co daný řádek popisuje, vypovídá až jeho struktura.

Oproti PDB jsou v tomto formátu zapsány první souřadnice daného atomu a až potom jeho chemické označení. Odlišností je zde i úvodní výčet celkového počtu molekul a vazeb. Popis vazeb je zde však totožný jako u formátu PDB. Více informací o tomto formátu lze nalézt zde [3].

```

3  2  0  0  0  0  0  0  0  0  V2000
0.0000  0.0000  0.1402  O  00000000000000000000
0.0000 -0.7678 -0.4207  H  00000000000000000000
0.0000  0.7678 -0.4207  H  00000000000000000000
1  2  1  0  0  0
1  3  1  0  0  0

```

Obrázek 2.3: Molekula vody zapsaná ve formátu MDL

Formát MDL může být také rozšířen o další data popisující chemické složení nebo vlastnosti dané molekuly. Toho lze dosáhnout využitím SDF, které taktéž spadá pod specifikaci Molfile. Soubor má totožnou strukturu jako MDL, ale v jeho závěru jsou ještě přidány SDF informace.

2.4 XYZ formát

Jedná se o jeden z nejvyžívanějších formátů pro popisování geometrie molekul. Hlavní výhodou je jeho jednoduchost, díky níž je využit ve většině editorů molekul. Na prvním řádku tohoto souboru se nachází počet atomů, na druhém je titulek, jímž je většinou název dané molekuly. Následuje popis jednotlivých atomů. Vždy na první pozici řádku je chemická značka daného atomu a následně jednotlivé koordináty z kartézského systému. Velkou nevýhodou je, že tento formát ve svém základu nedefinuje žádné vazby mezi jednotlivými atomy. To lze vyřešit jeho rozšířením, což však již není standardní a může být pro každý program pracující s tímto formátem různé.

```

3
# H2O.XYZ file for H2O
O  0.008165 -0.658276  0.215737
H  1.508729  0.441434  0.259102
H -1.419465  0.543189  0.243811

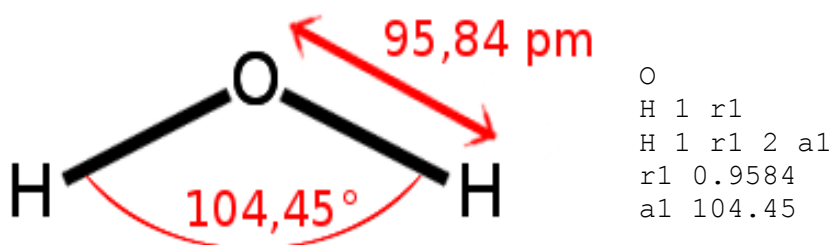
```

Obrázek 2.4: Molekula vody ve formátu xyz

2.5 Další formáty

Výčet formátů, jež jsou popsány v předchozích kapitolách, není ani zdaleka konečný. Existuje velké množství dalších formátů, které mohou být naprosto unikátní a specifické jen pro určitý program, nebo se může jednat o různé variace jednoho systému popisu geometrie molekul. Většina programů totiž pracuje s více formáty, z nichž většina patří do již zavedených a jeden je jejich vlastní. Formát vlastní danému programu se však většinou odlišuje hlavně v popisech nebo uspořádání souboru. Základ zůstává stejný. Je zapotřebí vyjádřit pozici jednotlivých atomů a případně i systém vazeb. Zatímco u vazeb se většinou pouze zapíše, které dva atomy k sobě pojí, u samotných atomů je třeba nějakým způsobem vyjádřit jejich pozici.

Většinou se využívá vyjádření pomocí souřadnic z kartézského systému, ale může se ojediněle vyskytnout i vyjádření pomocí tzv. z-matice. V takovémto případě je celá geometrie molekuly popsána nikoli pomocí souřadnic jednotlivých atomů, ale na základě jejich vzájemné pozice definované vzdálenostmi a úhly, které spolu navzájem svírají. Také se liší počet dat nutných u jednotlivých atomů pro popis jejich pozice. První atom neobsahuje žádné informace o jeho pozici, protože je umístěn v počátku tohoto systému. U druhého je již zapotřebí definovat jeho vzdálenost od prvního. Pro třetí atom je již zapotřebí i úhel, jež svírá s druhým z pohledu prvního atomu a samozřejmě také vzdálenost od prvního. Případné další atomy obsahují všechny již zmíněná data a navíc dihedralní úhel, jenž vyjadřuje úhel mezi jednotlivými rovinami.



Obrázek 2.5: Geometrie molekuly vody dle z-matice

2.6 Vzájemný převod formátů

Jelikož ne všechny formáty jsou podporovány všemi programy, je někdy zapotřebí formáty mezi sebou převádět. Díky relativně jednoduchému zápisu by bylo možno převod provést ručně. V případě neznámého formátu by bylo nutno toto provést například úpravou některého již existujícího souboru pro jinou molekulu. Pokud by se však jednalo o dva zcela odlišné formáty,

nebo o nějakou složitější molekulu, bylo by to již značně obtížné, pro některé uživatele až neproveditelné. V takovémto případě lze využít software určený k převodu formátů. Vzájemný převod největšího množství formátů podporuje program Open Babel. Tento program je zcela zdarma a je dostupný pro operační systémy Windows, Linux i Macintosh, přičemž umí pracovat s téměř všemi dostupnými formáty popisujícími molekuly. Více informací o programu Open Babel lze získat zde [4].

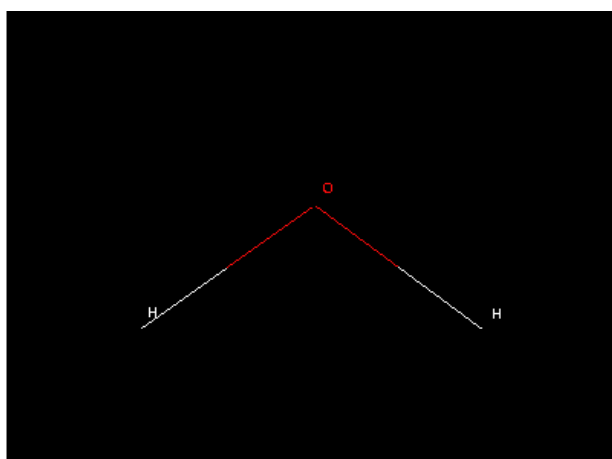
3 Grafické znázornění molekul

Stejně jako při počítačovém zobrazování čehokoli jiného, i u molekul je možno využít dvourozměrné nebo třírozměrné zobrazování. V praxi se tyto dvě zobrazení velice liší. 3D vyobrazení se zabývá hlavně skutečnou podobou dané molekuly a pozic jednotlivých jejích prvků, zatímco 2D znázorňuje jejich strukturu, avšak v některých případech se i tato zobrazení snaží zachytit také polohu prvků. Rozšířenější je dvourozměrné zobrazování, protože je mnohem jednodušší a většinou vychází pouze z chemického vzorce. Lze ho tedy relativně snadno a rychle ručně vytvořit a to bez jakýchkoli dalších informací, které jsou jinak obsaženy v souboru popisujícím geometrii molekulu. Pro zpracování výpočetní technikou a k tomu určenými programy se většinou využívá 3D zobrazování, zatímco 2D je hlavně pro ruční zpracování a například i pro školní účely.

3.1 3D modely molekul

3.1.1 Drátěný model

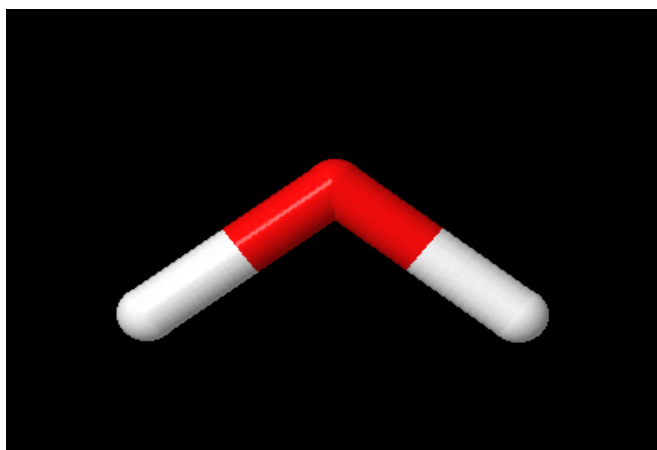
Dle názvu tohoto formátu by bylo možno si představit klasický drátěný model využívaný různými 3D programy pro zobrazování objektů. V tomto případě se však nejedná o klasické znázornění obrysů jednotlivých prvků, což by byly atomy a vazby, ale o pouhé znázornění každé vazby jedinou přímkou. Atomy se nacházejí ve spojích těchto vazeb a mohou být označeny jejich chemickou značkou. Každá vazba je barevně znázorněna dle toho, k jakému atomu se váže, přičemž pokud pojí dva různé atomy, je její označení jiné na obou polovinách.



Obrázek 3.1: Drátěný model molekuly vody – převzato z: [8]

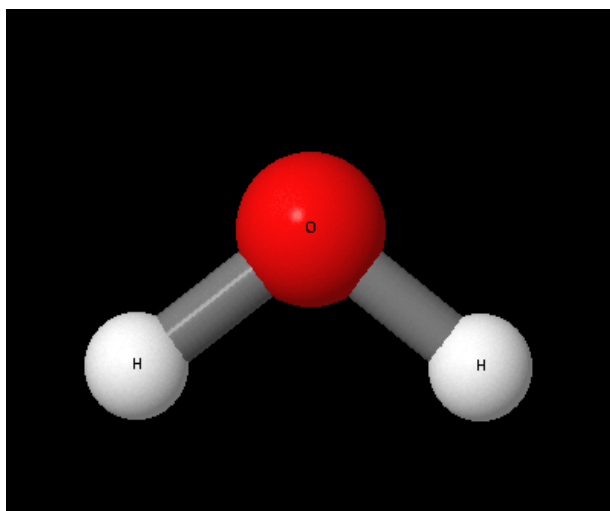
3.1.2 Tyčinkový model

Tyčinkový model je velice podobný drátěnému modelu. Vazby jsou zde již znázorněny poněkud lépe rozeznatelnými objekty ve tvaru tyčinek. Princip jejich obarvování však zůstává naprosto stejný jako u předchozího modelu. Stejně tak je možno jejich spoje, což jsou místa, kde se nachází jednotlivé atomy, opět opatřit popisky ve formátu jejich chemické značky.



Obrázek 3.2: Tyčinkový model molekuly vody – převzato z: [8]

3.1.3 Kuličkový model

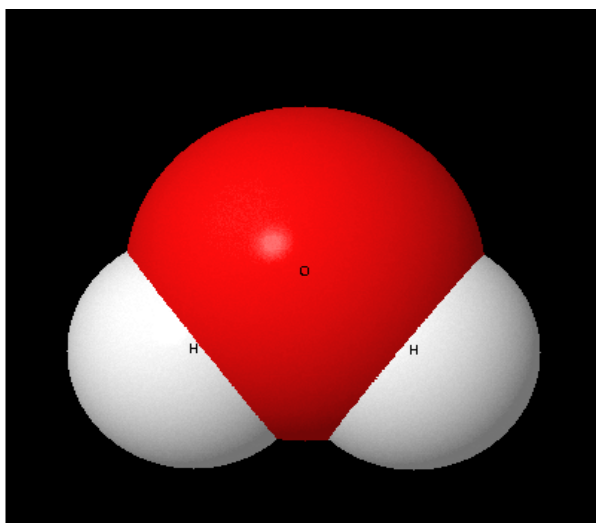


Obrázek 3.3: Kuličkový model molekuly vody – převzato z: [8]

Jedná se o nejčastěji využívaný a také pro malé až středně velké molekuly nejpřehlednější model. Jednotlivé atomy jsou zde znázorněny kuličkou obarvenou dle toho, o jaký prvek se jedná. Kuličku je možno opět ještě označit chemickou značkou prvku, jež znázorňuje. Kuličky jsou vzájemně spojeny válcí vyobrazujícími vazby mezi jednotlivými atomy. V některých případech se vykreslují pouze v jedné barvě, jež neodpovídá žádnému chemickému prvku, ale častěji se barví stejně, jako tomu bylo u předchozích popsaných modelů. Kuličkový model je možno přirovnat k tyčinkovému modelu s tím, že v místech, kde se protínají jednotlivé vazby, jsou vloženy patřičně obarvené kuličky, jež zvýrazňují skutečnost, že v daném místě se nachází atom.

3.1.4 Kalotový model

U tohoto typu modelu jsou znázorněny jednotlivé atomy opět jako kuličky, jejich vzájemná velikost je však určena počtem vazeb, jež z daného atomu vycházejí. Vazby jsou zde patrné pouze průnikem jednotlivých atomů. Při zobrazení větších molekul může být takovýto model značně nepřehledný.



Obrázek 3.4: Kalotový model molekuly vody – převzato z: [8]

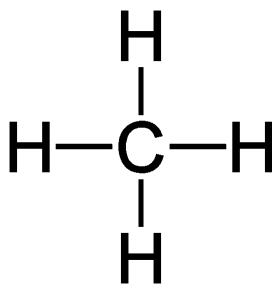
3.2 2D modely znázorňování molekul

Dvourozměrné vyobrazení molekul je realizováno pomocí jejich vzorců. Kromě základních, jež informují pouze o atomech v dané molekule a slouží čistě pro označení molekuly, existuje několik dalších typů, které již zachycují strukturu molekuly, případně i rozložení jednotlivých jejích atomů. Takovéto vzorce se nazývají strukturními. Oproti trojrozměrnému zobrazení se zde nevyužívá žádného barvení jednotlivých atomů. Ty jsou totiž vyjadřovány pouze jejich chemickou značkou a vazby úsečkou.

Tyto vzorce je možno rozdělit do kategorií a to na základě dvou kritérií. Jako první je možné dělení dle toho, zda vzorec zachycuje reálné pozice prvků v molekule, nebo se zaměřuje pouze na jejich vzájemné vazby. Druhým parametrem je to, jestli jsou vyobrazovány všechny atomy a to tak, že každý je zaznačen zvlášť, či mohou být ve vzorci místo atomů různé sloučeniny a některé prvky i vynechány, takže by daný vzorec obsahoval jen určité základní prvky.

3.2.1 Rozvinutý konstituční vzorec

Tento vzorec popisuje celou molekulu na základě její struktury, přičemž znázorňuje veškeré jednotlivé atomy a jejich vzájemné vazby. Tento vzorec se nezabývá skutečnou polohou atomů v dané molekule. Jelikož obsahuje veškeré prvky z dané molekuly, je jeho přehlednost dána velikostí této molekuly. V případě větších může dojít pouze ke snížení přehlednosti, ale také ke znemožnění zápisu molekuly tímto vzorcem.



Obrázek 3.5: Rozvinutý konstituční vzorec metanu

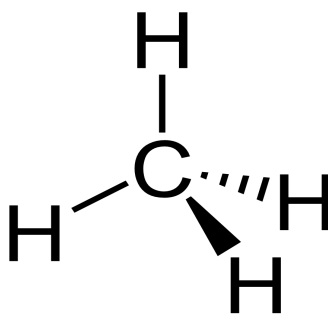
3.2.2 Racionální konstituční vzorec

Jde o speciální vzorec, vycházející z rozvinutého konstitučního vzorce. Zatímco ten vyobrazoval naprosto všechny atomy dané molekuly, zde se již znázorňují pouze atomy uhlíku nebo skupiny prvků, jejichž struktura je již dobře známa, proto není nutno ji celou rozkreslovat.

Při samotném zápisu tedy tento vzorec zabere mnohem méně místa, ale porozumění jeho obsahu již vyžaduje více chemických znalostí.

3.2.3 Geometrický vzorec

Geometrický vzorec se snaží co nejvěrněji zachytit skutečnou polohu všech atomů ve znázorňované molekule. U prvků, jež v dané molekule leží v jedné rovině, je to značně jednoduché. Problém se zobrazením však nastává tehdy, kdy prvek leží mimo ni. V tomto případě je zapotřebí rozlišit i to, jestli leží před nebo za touto rovinou (zda leží v popředí nebo v pozadí). Toho se většinou docíluje tím, že vazba není zakreslena obyčejnou úsečkou, ale jakýmsi klínem. Pokud je prvek v popředí, tak je tento klín vyobrazen plnou čarou. V případě, že leží v pozadí, bývá vazba znázorněna přerušovanou čarou.



Obrázek 3.6: Geometrický vzorec molekuly metanu

4 Aplikace pro 3D vizualizaci molekul

4.1 Rasmol

Tento program je primárně určen k vizualizaci molekul, kyselin a proteinů. Pracuje s několika formáty popisujícími geometrii molekul. Jedná se o formát Mol, MDL, PDB, XYZ, CIF a mmCIF. Výsledná vizualizace může být následně vyexportována např. ve formátu BMP nebo GIF.

Molekuly lze zobrazovat jako drátěný, kuličkový nebo také kalotový model. Barvu atomů lze nastavit dle více barevných schémat. U všech jsou však barvy jednotlivých atomů určeny prvkem daného atomu. Pro každý prvek také umožňuje zobrazit jeho název.

Jedná se o otevřený software, který je zdarma ke stažení přímo na stránkách tvůrců. Je dostupný pro většinu nejvyužívanějších operačních systémů, konkrétně pro Windows, Macintosh a Linux. Více informací o tomto programu lze nalézt zde [5].

4.2 Molegro Molecular Viewer

Primární funkcí tohoto programu bylo původně zobrazování výsledků z molekulárního programu Molegro Virtual Docker od stejného výrobce, který slouží k simulaci proteinů. Molegro Molecular Viewer však kromě toho také slouží k vizualizaci molekul zapsaných ve formátech PDB, Mol2 a SDF. Umožňuje také do těchto formátů exportovat.

Stejně jako Rasmol i tento program je zcela zdarma a dostupný pro různé operační systémy. Program je možno využívat na operačních systémech Windows, Linux a Macintosh. Více informací o programu Molegro Molecular Viewer je k dispozici zde [6].

4.3 Jmol

Jmol je open source program pro 3D vizualizaci molekul napsaný v jazyce Java. Jmol je implementován v několika verzích, díky čemuž dokáže pracovat jako samostatný program, jako integrovaná část v jiném programu nebo jako Java applet. Jeho implementace je použitelná na operačních systémech Windows, Linux i Macintosh, přičemž existuje i verze pro operační systémy Android a iOS.

Ve všech verzích je při vizualizaci možno volit mezi jednotlivými modely zobrazení dané molekuly (drátěný, kuličkový, kalotový) a libovolně rotovat kameru. Jeho ovládání je primárně založeno na textovém vstupu, kdy se jednotlivé příkazy přímo zapisují a program je následně vykonává. Obsahuje také možnost přímého přístupu do databáze PDB a NIH/NCI.

Jmol podporuje mnoho formátů, přičemž jeho rozsah je možno přirovnat k programu Open Babel, jenž slouží právě ke vzájemnému převodu jednotlivých formátů. Další informace o programu Jmol jsou dostupné zde [8].

5 Možnosti 3D vizualizace na mobilních platformách

5.1 JavaME

Ve své základní podobě Java Micro Edition neobsahuje žádná API pro podporu a práci s 3D grafikou. Tento nedostatek byl výrobci mobilních zařízení řešen pomocí svých vlastních knihoven, jež byly implementovány pro jejich zařízení a také v nich využity. To však omezilo použitelnost aplikací, jež tato API využívaly, pouze na zařízení, pro něž byly tyto aplikace přímo napsány (zařízení jednoho výrobce obsahující 3D knihovnu). Většina těchto knihoven pracovala pouze s celými čísly.

Jednotné 3D API vzniklo až v roce 2003 mezinárodní organizací vývojářů. Jedná se o JSR-184. Stále je to pouze volitelné API, které však již je obsaženo ve většině nových mobilních zařízeních. Oproti původním knihovnám od výrobců zařízení je standardizováno a je optimalizováno tak, aby bylo co nejméně náročné na výkon a paměť vlastního zařízení. Také již podporuje práci s desetinnými čísly.

Stejně jako u 2D zobrazování i zde je obsažena třída Canvas sloužící jako scéna. Také vykreslování scény je řešeno pomocí metody paint. Jednotlivé objekty v ní jsou adresovány pomocí jedinečných identifikátorů, jež jsou jim přiděleny při jejich načtení. Jejich pozice je zapsána pomocí trojrozměrného souřadnicového systému. Jedním z objektů, jež mohou být do scény vloženy je objekt sloužící jako kamera. Všechny objekty včetně kamery mohou být libovolně pozicovány, rotovány, přidávány i odebírány.

Nejnovější knihovnou pro 3D grafiku v rámci JavaME je JSR-239, jež zavádí podporu knihoven OpenGL pro 2D a 3D grafiku.

5.2 Windows Mobile, Windows Phone

Systémy z rodiny Windows využívají k vykreslování 3D grafiky pomocí souboru tříd Direct3D Mobile. Ty jsou optimalizovány pro běh na zařízeních s minimální pamětí a výkonem. Podporují práci s fixní plovoucí i desetinnou čárkou, pro snížení paměťové náročnosti. Mobilní zařízení podporují oba formáty a dle aktuálního stavu (výkonové a paměťové náročnosti) volí, který typ se pro daný případ využije.

U Windows Phone je vývoj 3D grafiky řešen pomocí rozhraní XNA, jež bylo vyvinuto za účelem tvorby 3D her. Stále však využívá DirectX a je multiplatformní. Umožňuje vývoj i pro Xbox či běžné počítače.

5.3 Android

Android obsahuje pro práci s grafikou základní třídy velice podobné těm z JavaME. Lze je použít k tvorbě uživatelského rozhraní nebo vykreslování grafiky. Slouží však jen pro práci s 2D grafikou.

Již od verze 1.0 Android obsahuje vlastní OpenGL ES API s podporou knihovny OpenGL pro práci s 2D a 3D grafikou. To je velice podobné JSR-239 používané u platformy JavaME. OpenGL je podporováno jak na úrovni SDK, tak i prostřednictvím NDK.

OpenGL využívá k určování pozice čtvercový souřadnicový systém, zatímco zařízení se systémem Android mají většinou různě velké displeje a poměr jejich stran. Pro každé zařízení však zůstane původní souřadnicový systém stejný, ale provede se transformace, která upraví velikost jednotlivých čtverců tak, aby pokryly potřebnou plochu displeje.

6 Analýza programu

6.1 Výběr platformy

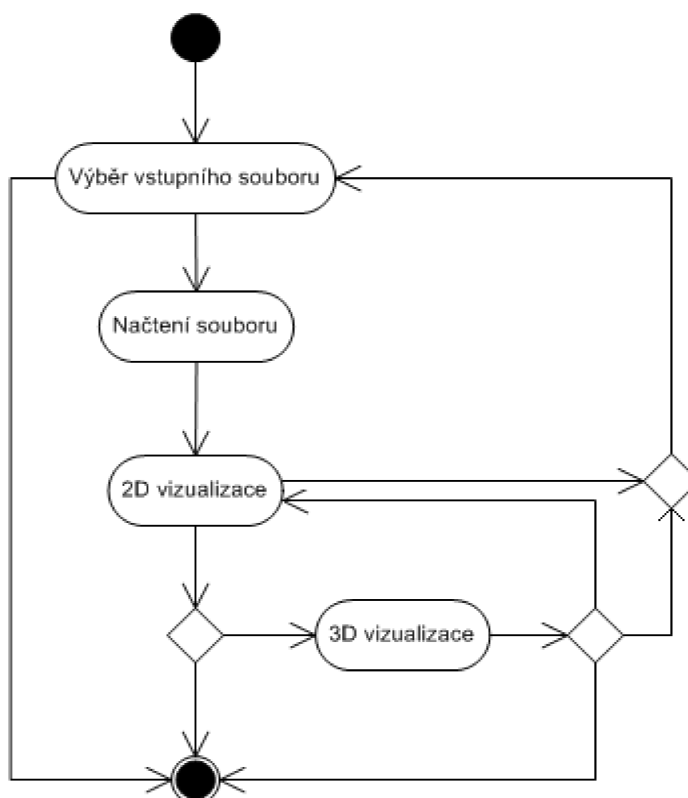
Pro implementaci programu bude využito platformy Android. Podíl mobilních telefonů s operačními systémy mezi uživateli každoročně roste, přičemž nejvyšší nárůst zaznamenávají právě zařízení se systémem Android. Dle analýz společností IDC a ABI Research by dokonce mohl Android dosáhnout do čtyř let 45% zastoupení na trhu.

Veškerá zařízení s touto platformou jsou výkonově i programově schopná pokročilého zobrazování 3D grafiky. Rozměry a rozlišení displejů také přispívají k možnosti přehledné vizualizaci větších molekul obsahující vyšší počet atomů.

6.2 Základní funkce programu

Aplikace bude schopna načíst strukturu molekuly ze vstupního souboru a dle výběru uživatele ji vykreslit ve 2D a 3D prostředí. Vstupní soubor bude v jednom z implementovaných existujících formátů popisujících geometrii molekul. Z něj se vyberou jednotlivé atomy a určí se jejich pozice v kartézském souřadnicovém systému a jejich chemická značka. Rovněž se, u formátů, jež podporují jejich zápis, dle vstupního souboru určí pozice všech jednotlivých vazeb. Všechny takto uložené objekty se vykreslí. Uživatel bude mít možnost volného pohybu kamery, zahrnující přibližování, oddalování a rotaci. Během 3D zobrazení molekuly bude také moci zvolit změnu modelu zobrazení molekuly.

Vždy při spuštění aplikace uživatel vybere pomocí jednoduchého souborového prohlížeče vstupní soubor pro načtení molekuly, jež si přeje vizualizovat. V prohlížeči budou zobrazeny všechny soubory nacházející se v daném adresáři. Aplikace postoupí na další krok až po výběru souboru s patřičnou příponou, tedy po výběru podporovaného souboru. Následuje načtení molekuly ze zvoleného souboru. Aplikace dle přípony rozliší, o jaký formát se jedná a na základě toho zvolí patřičný systém převodu. Načtením souboru se získají dva seznamy, z nichž jeden obsahuje všechny definované atomy a druhý vazby. Tyto seznamy slouží k následné vizualizaci těchto prvků. Před samotnou vizualizací se oba seznamy projdou a provedou se výpočty potřebné pro optimální vykreslení. Druhý průchod následuje až při vykreslování.



Obrázek 6.1: Základní struktura aplikace

6.3 Import souborů

Dle formátu vstupního souboru bude vytvořena třída, která bude provádět a řídit celý import. Všechny tyto třídy budou implementovat stejné rozhraní, což zajistí stejný způsob vstupů a výstupů pro soubor jakéhokoli podporovaného formátu.

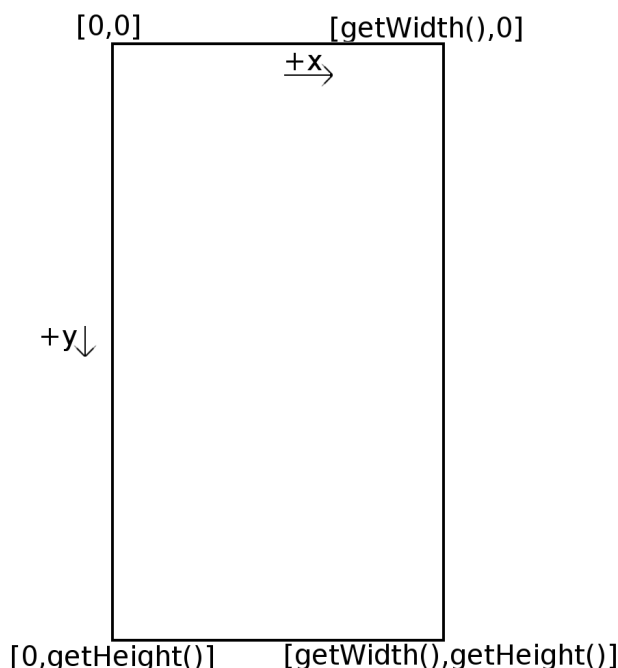
Kromě využití stejného rozhraní budou využity pro zápis získaných dat vždy stejné prvky. Konkrétně se bude jednat o dvě třídy, které budou reprezentovat základní stavební prvky samotné molekuly a to atomy a vazby. Pomocí nich se zaznamenají základní vlastnosti všech načtených objektů.

Samotné načítání daného souboru bude prováděno dle jeho formátu, přičemž aplikace bude ve své základní podobě umět načíst soubory XYZ, PDB a CML. Pro XYZ a PDB bude nutno

implementovat celý systém řízení načítání, zatímco u CML, jež vychází z XML struktury, je možné využít stávajícího XML parseru a pouze definovat jednotlivé potřebné události.

6.4 2D vizualizace

Během 2D zobrazení bude viditelná celá molekula, což poskytne uživateli přehled o její celkové struktuře. Jednotlivé atomy budou obarveny dle implementovaného barevného schématu. Pokud je molekula ve vstupním souboru zapsána s trojrozměrnými souřadnicemi, bude souřadnice z ignorována. Tímto může zvláště u větších molekul dojít ke vzájemnému překrytí jednotlivých prvků, čemuž se však u rozměrných molekul nelze žádným způsobem vyhnout.



Obrázek 6.2: Souřadnicový systém displeje pro třídu Canvas

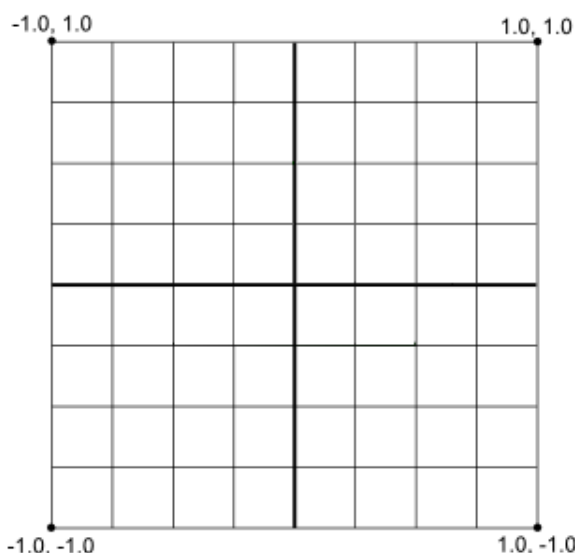
Pro vykreslování 2D grafiky bude využito standardní třídy Canvas, jež umožňuje jednoduché kreslení 2D grafiky. Canvas obsahuje přímé kreslení geometrických objektů a adresování polohy přímo dle jednotlivých obrazových bodů na displeji. Počátek tohoto souřadného systému je v bodě $[0,0]$, který je umístěn v levém horním rohu. Souřadnice na ose x stejně jako u kartézského souřadnicového systému směrem do prava od počátku rostou, zatímco na ose y rostou ve směru dolů od bodu počátku. Pro ještě snazší a efektivnější pozicování kreslených

objektů lze využít metod, jež vrací výšku a šířku displeje, tedy dohromady souřadnice pravého dolního rohu. Souřadnicový systém displeje pro kreslení pomocí třídy canvas je znázorněn na obrázku 6.2.

Samotnou molekulu je potřeba do tohoto systému umístit tak, aby zabrala pokud možno co největší plochu zobrazovací jednotky a minimalizovalo se tím nevyužité místo. Souřadnice všech prvků připravených pro vykreslení bude nutno převést do souřadnicového systému daného třídou canvas a velikostí displeje. Během převodu se zvolí poměr a posunutí. Poměr se zvolí na základě rozsahu souřadnic na osách x a y dané molekuly a také velikosti displeje snížené o 40 bodů, což zajistí, že se molekula nebude dotýkat okraje displeje ani z něj na okrajích vyčínat. Pro obě osy využívané ve 2D zobrazení se vypočte násobitel a následně se zvolí ten nižší, čímž bude zajištěno, že molekula se vejde na displej celá. Molekula pak bude nutno posunout tak, byl její levý horní roh v bodě [0,0]. Podle její výšky a šířky se vypočte volné místo pod ní a napravo od ní, aby bylo možno ji posunout do středu displeje.

6.5 3D vizualizace

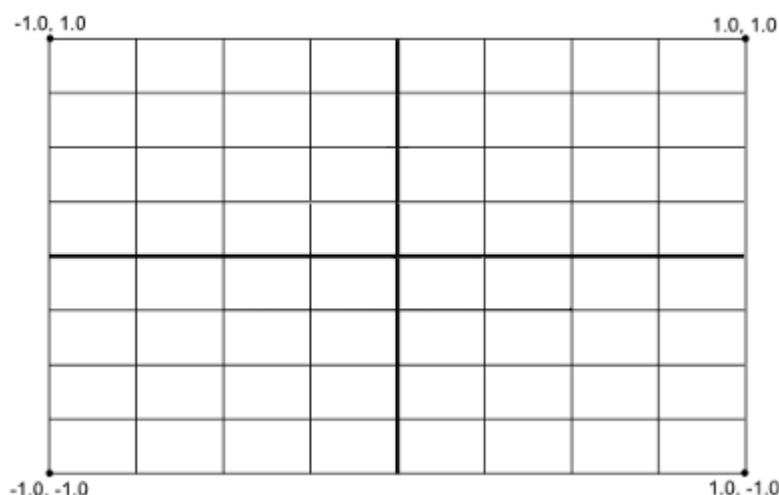
Stejně jako při dvourozměrném zobrazení, tak i při 3D vizualizaci bude platit, že molekula bude na displeji zobrazena celá. Vzhledem k tomu, že je nutno zde počítat i s osou z, která je orientována kolmo na plochu displeje, bude nutno provést veškeré výpočty i pro ni.



Obrázek 6.3: Základní souřadnicový systém OpenGL

Molekulu bude možno rotovat kolem jejího středu, jež bude vždy umístěn uprostřed displeje. Celkový prostor, jež bude molekula při jakékoli rotaci zabírat, je možno si představit jako kouli s průměrem odpovídajícím velikosti molekuly v jejím nejširším místě.

OpenGL ES, jež bude využito pro 3D znázornění, využívá zcela odlišný souřadnicový systém než tomu bylo u vykreslování grafiky pomocí třídy Canvas. Pro jednoduchost se zde počátek, což znamená bod $[0,0]$, nachází uprostřed systému a také uprostřed displeje. Rozsah systému na všech třech osách je od -1 do 1. Čtvercový tvar celého souřadnicového systému lze však v případě potřeby transformovat a kompletně jím pokrýt zobrazovací jednotku o jakémkoli poměru stran. Společně se změnou velikosti os, které si stále zachovají rozsah adresování od -1 do 1, však dojde také k deformaci objektu, jež je zde umístěn. Tu je možno dále opravit pomocí projekční matice, která souřadnice vykresleného objektu přepočte pro nový poměr os.

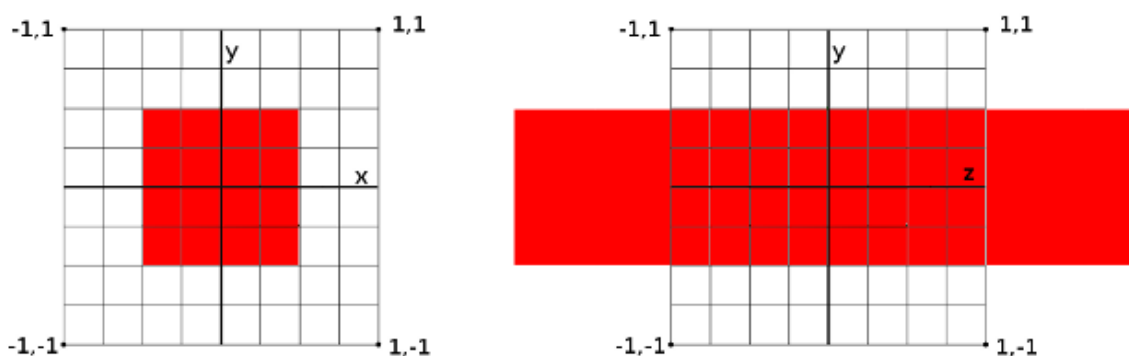


Obrázek 6.4: Transformovaný souřadnicový systém OpenGL

Celkovou velikost před použitím přiblížení bude určovat opět násobitel. Princip jeho určení bude totožný jako u 2D vizualizace, ale bude se tentokrát vypočítávat pro všechny tři osy a na závěr se opět zvolí ten nejnižší. V případě volby jiného, by se molekula při některých rotacích zobrazovala mimo vykreslovací oblast displeje.

Na obrázku 6.5 je znázorněno vykreslení kvádra s délkou hran $\{1,1,4\}$ do souřadného systému využívaného OpenGL se špatnou volbou poměru. Objekt má být dostatečně vzdálen od okrajů systému, jehož šířka a výška je rovna 2. Nyní nevyberem nejdelší stranu, jak by tomu správně mělo být, ale zvolíme hranu a , která má délku jedna. Pokud objekt umístíme do středu zobrazovací plochy a zvolíme násobitel 1, zůstane nám tak na jejích okrajích prostor o velikosti 0,5. Objekt je tedy zobrazen celý. Jakmile se však provede rotace o 90° kolem osy y , je již

z větší části mimo zobrazovací oblast. Při správném poměru sice bude při většině rotací zobrazovaný objekt zabírat menší část displeje, ale při žádné poloze os se nedostane mimo zobrazovací oblast.



Obrázek 6.5: Špatná volba poměru objektu při 3D zobrazení

Kamera bude stabilně umístěna na jednu pozici s možností pohybu dle osy z , což znamená pohyb dopředu (přiblížení) a pohyb dozadu zpět na počáteční pozici (oddálení). Rotace bude řešena rotováním celé molekuly. Druhou variantou by byl skutečný pohyb kamery po kružnicích kolem celého objektu. Tato varianta je však zbytečně složitější. Pro rotaci celého objektu stačí vždy před jeho vykreslením pootočit celý systém os bez nutnosti jakékoli dodatečné implementace přepočtu polohy objektů. Přibližování a oddalování kamery bude ovládáno vzájemnou polohou dvou prstů na displeji. Pohyb prstů k sobě bude molekulu přibližovat zatímco pohyb od sebe oddalovat. Toto ovládání se využívá standardně i v mnoha jiných aplikacích pro zařízení podporující detekci více míst dotyku najednou (tzv. multitouch).

Pro dosažení vizuálního 3D efektu bude nutno do implementované scény umístit také zdroj světla. Bez něj by výsledný obraz vypadal pouze jako při 2D zobrazení a jeho 3D vlastnost by byla patrná až při rotaci molekuly.

3D vizualizace bude obsahovat tři základní modely zobrazování molekul. Konkrétně se bude jednat o kuličkový model, jenž bude primární a zobrazí se vždy prvně po spuštění 3D zobrazení, kalotový a drátěný model. Dle zvoleného modelu se budou lišit prvky, jež bude zapotřebí vykreslovat. V každém z těchto modelů budou odlišné. U kuličkového budou vykreslovány atomy i vazby mezi nimi. Kalotový model bude znázorňovat pouze atomy a drátěný naopak pouze vazby.

6.6 Přepínání režimů vizualizace

Během obou typů vizualizace bude nutno umožnit jak přechod zpět na výběr vstupního souboru, tak i přepnutí mezi nimi navzájem. Přechod na výběr vstupního souboru bude implementován pomocí tlačítka zpět a bude shodný pro 3D i 2D vizualizaci.

Pro přepínání mezi režimy zobrazování bude sloužit menu, jehož položky se budou měnit vždy při změně typu vizualizace. Pokud by zůstávaly stejné, byla by například při 2D vizualizaci nejen možnost přepnout na 3D vizualizaci, ale také nesmyslná položka na přepnutí do 2D vizualizace, v níž by se aktuálně uživatel stejně nacházel.

Obměňování položek v menu umožní i jeho další využití v režimu 3D vizualizace. Jelikož tento režim bude mít implementovány 3 zobrazovací modely molekul, bude zapotřebí nějakým způsobem zajistit jejich přepínání. Vhodnou metodou je právě umístění volby těchto modelů přímo do menu. To se zamožřejmě provede pouze při změně režimu na 3D vizualizaci.

7 Implementace

7.1 Základní struktura aplikace

Celá aplikace se skládá ze dvou aktivit. První z nich je zaměřena pouze na výběr vstupního souboru, což znamená, že obsahuje jen jednoduchý souborový prohlížeč. Ten je implementován pomocí standardního ListView, jež je jednou z předdefinovaných komponent, které ve svém základnu poskytuje Android SDK pro tvorbu grafického uživatelského rozhraní. Po úspěšném výběru je připravena druhá aktivita, která má již na starosti vizualizaci.

Před samotným spuštěním druhé aktivity je však zapotřebí jí předat vstupní soubor, na jehož základě má být vizualizace prováděna. Nová aktivita je vytvořena pomocí objektu typu Intent. Ten umožňuje také předávat proměnné mezi jednotlivými aktivitami a to metodou, kdy se zvolí ukládaný název proměnné včetně balíku, do nějž náleží, a pak i její vlastní hodnota. Po spuštění nové aktivity je možno si takto uloženou proměnnou z předchozí aktivity v té aktuální vyzvednout.

Ve druhé aktivitě se předaný vstupní soubor importuje a dle něj se zde provádí i vykreslování. Standardní vlastností operačního systému Android je, že při stisku klávesy zpět se vždy vrací na minulou neukončenou aktivitu. Díky tomu je zajištěn zpětný vstup z vizualizace na výběr nového vstupního souboru bez potřeby jakékoli dodatečné implementace této vlastnosti.

7.2 Souborový prohlížeč

Jednoduchý souborový prohlížeč slouží k výběru vstupního souboru popisujícího molekulu, který se nachází na paměťové kartě daného zařízení a jež si přeje uživatel zobrazit. Jeho grafické rozhraní je realizováno pomocí grafické komponenty ListView.

Vždy při prvním spuštění je načten seznam všech adresářů a souborů, jež se nacházejí v kořenovém adresáři paměťové karty. Na první místo je navíc vždy vložena položka sloužící k přechodu o adresář výše. Tento seznam je nadále přiřazen k objektu typu ArrayAdapter, který umožňuje propojení s připraveným grafickým objektem znázorňujícím jednu položku. Grafické komponentě ListView je následně přiřazen takto připravený adaptér, čímž dojde k namapování seznamu souborů na jednotlivé položky(řádky) v grafickém seznamu. Zároveň je nastavena hodnota objektu TextView, který slouží jako ukazatel aktuální pozice na paměťové kartě zařízení.

Celému grafickému seznamu je přiřazen OnItemClickListener. Ten má za úkol identifikovat případný dotyk uživatele a rozpoznat, která položka byla takto zvolena. Při zaznamenání doteku se jako první ověří, zda nebyla zvolena první položka v seznamu. Pokud ano, je ověřeno, zda je

možno postoupit v adresářové struktuře o adresář výše či nikoli. V případě kladné možnosti se používaný seznam položek vyprázdní a načtou se do něj nové, které se nachází ve struktuře o úroveň výše. Změnou seznamu položek dojde i k obnovení celého grafického seznamu (ListView). Také se změní hodnota(text) ukazující aktuální pozici na paměťové kartě.

Pokud není zvolen přechod o úroveň výše(první položka v seznamu), ověří se, zda byl vybrán soubor nebo adresář. V případě adresáře se provedou podobné kroky jako v předchozím případě, avšak není nutno kontrolovat zda je tento krok možný.

Jakmile je vybrán soubor, ověří se, jestli je jeho přípona shodná s některým z podporovaných vstupních formátů. V kladném případě je připravena nová aktivita ,která se bude starat o import a vykreslení na základě vstupního souboru.

7.3 Atomy a vazby

Při importu vstupního souboru jsou vždy vytvářeny seznamy obsahující jednotlivé základní stavební prvky celé molekuly. Tyto objekty obsahují základní parametry, které je možno získat z importovaného souboru a pak i dodatečné, které je možno jim přiřadit později pro usnadnění jejich vizualizace.

7.3.1 Třída Atom

Tato třída slouží pro reprezentaci jednoho základního stavebního prvku, konkrétně atomu. Jejím základním údajem je jeho poloha a chemická značka prvku, jemuž daný atom náleží. Pro určení polohy atomu slouží tři proměnné, z nichž každá vyjadřuje pozici na jedné ze tří os. Pokud by molekula ve vstupním souboru byla zapsána pouze ve 2D systému, tak je nutno při vytváření atomu nastavit pozici na ose z na hodnotu 0.

Atom také obsahuje proměnnou pro uložení identifikátoru, který může být u některých formátů potřebný k určování poloh vazeb, jež z daného atomu vycházejí. Celkový počet vazeb tohoto atomu je také třeba zaznamenat pro pozdější potřeby 3D zobrazení.

Kromě těchto parametrů obsahuje objekt třídy Atom ještě položku obsahující údaj o jeho barvě. Ta se však zaznamenává až později před samotným vykreslením na základě prvku daného atomu. Tato informace pak obsahuje pouze číslo, jež určuje, na které pozici se v celé škále barev tato konkrétní nachází.

7.3.2 Třída Bond

Třída Bond slouží k reprezentaci vazeb mezi jednotlivými atomy. Zatímco u atomů stačí udržovat informace o jejich umístění pomocí jediného bodu, u vazeb je údaj o jejich poloze dán dvěma body. Ty jsou shodné s pozicemi prvního a druhého atomu. Kromě toho je zapotřebí uchovat i chemickou značku obou atomů, mezi nimiž se vazba nachází.

Stejně jako u třídy Atom i zde je implementován mechanismus pro uložení informace o barvě vazby. V tomto případě se však jedná o dvě proměnné, do kterých se těsně před samotným vykreslením uloží informace o barvě obou atomů ležících na jejich koncích.

Během vykreslování povedou všechny vazby od středu prvního atomu do středu druhého. To by však vylučovalo definici dvojných vazeb pomocí jejich dvojitého vložení do seznamu. Proto je u vazeb připraven i číselný parametr určující, zda se jedná o jednoduchou či dvojnou vazbou.

7.4 Importní třídy

Objekty obstarávající import vstupního souboru jsou vytvářeny až ve druhé aktivitě, přičemž se vytváří vždy jen jeden. O tom, který konkrétní objekt je zapotřebí vytvořit se rozhoduje na základě přípony dodaného vstupního souboru. Ten může být ve formátu XYZ, PDB nebo CML. Ve všech případech však objekt obstarávající import tohoto souboru implementuje jedno shodné rozhraní.

7.4.1 Společné rozhraní

Toto rozhraní slouží ke specifikaci základní stavby a funkčnosti všech objektů zajišťujících import vstupních souborů popisujících geometrii molekul. Obsahuje celkem šest různých metod.

První z nich slouží k samotnému zahájení převodu. Jako vstup jí slouží právě vstupní soubor dodaný do aktuální aktivity tou předchozí, jež obsahovala souborový prohlížeč. Další dvě metody slouží k přidávání nových atomů a vazeb do jejich celkových seznamů, které lze získat za pomoci dalších dvou specifikovaných metod.

Poslední metoda nemusí být nutně u všech importních tříd dopsána. Slouží totiž k přetypování jednotných vazeb na vazby dvojně. Konkrétně těch vazeb, jež mají stejné souřadnice obou atomů, které propojují. Některé formáty zápisu molekul tuto metodu potřebují, protože dvojně vazby jsou specifikovány pouze jejich zdvojeným uvedením v seznamu vazeb, zatímco u jiných, které tuto metodu nevyužijí, jsou dvojně vazby přímo ve svém zápisu označeny patřičným symbolem či znakem.

7.4.2 Formát XYZ

Import formátu XYZ je nejjednodušším implementovaným importovaným formátem. Jeho jednoduchost spočívá v tom, že v základní podobě neobsahuje definici vazeb, ale pouze atomů. Přesto je v něm definováno i pole pro vazby, protože nezávisle na jejich absenci bude procházeno při samotném vykreslování.

Čtení souboru probíhá po jednotlivých řádcích. Jakmile jsou všechny přečteny, navrací metoda pro započítání převodu hodnotu `true`. Během načítání molekuly se první dva řádky přeskóčí. Obsahují pouze celkový počet atomů a řádek určený pro komentář či název molekuly.

Každý atom je zapsán samostatně na jednom řádku. Po načtení je celý řádek rozložen na několik částí podle prázdných míst, jež slouží jako oddělovače. Celkem se tímto způsobem může řádek rozložit na tři až čtyři části dle toho, zda soubor obsahuje zápis molekuly ve 2D nebo 3D systému. První část řádku obsahuje chemickou značku atomu. Další tři, případně dvě určují jeho polohu. Na základě těchto informací se vytvoří nový Atom a metodou z implementovaného rozhraní se vloží do seznamu obsahujícího všechny dosud načtené atomy.

Při vytváření atomu se volí konstruktor bez určení jeho id, protože tento vstupní formát ho neobsahuje a ani by to nebylo vzhledem k absenci vazeb zapotřebí.

7.4.3 Formát PDB

Stejně jako u formátu XYZ je i zde údaj pro každý jednotlivý prvek molekuly zaznamenán na jednom řádku. Tím je způsobeno shodné využití čtení vstupního souboru po jednotlivých řádcích. Každý řádek se dále zpracovává dle toho, jaký prvek obsahuje.

Informace o tom, jakou součást molekuly daný řádek popisuje, je umístěna na prvních sedmi pozicích každého řádku. Stejně tak i ostatní informace jsou zapsány na přesně daných pozicích. Celý řádek je dále zpracováván pouze tehdy, je-li na jeho začátku řetězec `ATOM` nebo `CONNECT`, které označují zápis jednoho atomu či vazby.

Každý jednotlivý zapsaný parametr může být kratší, než je rozsah jeho umístění. V takovém případě jsou využity k zápisu pozice s vyšším adresou(pořadí znaku), zatímco zbytek celého prostoru je vyplněn mezerami. To znamená, že v celém rozsahu určeném pro jeden parametr jsou jeho hodnoty psány napravo, zatímco zbývající místo vlevo je prázdné. Při samotném čtení takto zapsaných hodnot je možno pomocí metody `trim` odstranit prázdné místo a získat řetězec obsahující pouze požadovaný parametr. Ten je dále zpracováván dle toho, jestli je potřebný jako číselná hodnota či text.

U atomů stačí pouze z celého řádku vybrat potřebné části, konkrétně se jedná o chemickou značku prvku atomu, jeho identifikátor a koordináty. Po získání těchto parametrů se vytváří objekt třídy Atom a vloží se do seznamu atomů.

Vazby mezi atomy jsou u tohoto vstupního formátu zapsány ve skupinách. Každá skupina obsahuje vazby začínající ve stejném atomu. Na začátku řádku je stejně jako u atomů zapsáno, o jaké části molekuly se jedná. Další pozice obsahuje identifikátor atomu, ve kterém vazby toho řádku začínají. Pro získání jeho pozice je nutné projít celý seznam atomů a dle id určit odpovídající atom. Tento atom se uloží do proměnné dokud se nedokončí zpracovávání celého řádku. Následující pozice obsahují identifikátory koncových atomů vazeb. Postupně se načtou a stejným způsobem jako u prvního bodu se i zde vybere ze seznamu správný atom. Pomocí informací z počátečního a aktuálně načteného atomu se vytvoří nový objekt třídy Bond a uloží do seznamu vazeb. Zároveň s tím je inkrementována proměnná, jež v atomu určuje počet jeho vazeb.

Jelikož dvojné vazby jsou v PDB formátu zapsány pomocí definice z počátečního i koncového bodu, je takováto vazba do seznamu vložena dvakrát. Při vizualizaci by se pak pouze překreslila jedna přes druhou. Z toho důvodu je doplněna metoda typeBond z implementovaného rozhraní. Ta postupně projde celý seznam a všechny vazby jsou porovnány. Pokud jsou nalezeny vazby s totožnými body, je ze seznamu jedna odstraněna a druhé změněn její typ.

7.4.4 Formát CML

Pro import toho formátu je využito SAX parseru, který zajišťuje načítání XML struktury souboru a hlásí načtení jednotlivých elementů. Samotný reaguje na všechny počáteční i koncové elementy s tím, že jako vstupní parametry pro metody reagující na tyto události slouží jsou vlastní načtené hodnoty. Lze z nich tedy rozeznat o jaký konkrétní element se jedná a dle toho dále zpracovávat informace o něm.

K získání všech potřebných dat pro vykreslení stačí načítat jen atomy a vazby, proto program kontroluje pouze to, zda se jedná o počáteční či koncový element označující jednotlivé atomy nebo vazby. Jestliže se jedná o atom, jsou vždy veškeré informace o něm obsaženy v samotném jeho elementu, tudíž jsou obsaženy ve vstupních parametrech metody identifikující začátek elementu. Pomocí se nich se vytvoří objekt třídy Atom a vloží do seznamu atomů.

Vazby bývají v některých případech definovány stejně jako jako atomy, tedy přímo v počátečním elementu. V takovém případě jsou stejně tak hned načteny všechny potřebné informace o nich. Jejich pozice je však tak jako u formátu PDB udána pomocí identifikátorů atomů, jež spojují. Proto jsou opět v seznamu atomů vyhledány příslušné dva atomy a s pomocí informací o jejich poloze vytvořen objekt třídy Bond.

V některých souborech mohou být vazby popsány vnořenými elementy typu string. Ty pak obsahují parametr, určující co který element konkrétně obsahuje. Mezi počátečním a koncovým elementem se pak nachází samotná konkrétní hodnota. Pro import i takto zapsaných vazeb je při zjištění počátku vazeb nastavena proměnná, která určuje, že se bude číst takto zapsaná vazba.

Samotné čtení hodnot je pak zajištěno metodou `characters`, kterou obsahuje SAX parser právě pro čtení samotných dat. Po načtení konce vazby se dohledají dle identifikátorů atomu, jež spojuje, a je vytvořen objekt třídy `Bond`. Ten se vloží do seznamu všech vazeb.

7.5 Příprava vizualizace

V tomto kroku se vytváří nová aktivita. Ta si pomocí metody `getExtra` vyzvedne předaný řetězec z předchozí aktivity (souborový prohlížeč), jež obsahuje adresu vstupního souboru. Na základě přípony vstupního souboru rozhodne, pomocí které třídy se bude provádět import molekuly. Všechny implementují stejné rozhraní, takže se po jejím vytvoření shodně volají metody pro spuštění převodu a získání seznamů atomů a vazeb. Na základě proměnné obsahující údaj, zda se má vizualizovat ve 2D nebo 3D se vytvoří nové `View`, jehož vstupními parametry jsou právě seznamy jednotlivých částí molekuly.

Kromě spouštění jednotlivých typů vizualizací se zde připravuje a obsluhuje i menu, jež se bude zobrazovat při samotných vizualizacích. Na základě proměnné ukazující aktuálně zvolený typ zobrazení se připraví položky nabídky. Pro 2D vizualizaci je to pouze položka pro přepnutí na 3D. U 3D se vkládají možnosti pro změnu na 2D, přepnutí zobrazovacího modelu molekuly a zapnutí automatické rotace modelu. Vždy při změně typu se celá nabídka vymaže a nově se do ní vloží potřebné možnosti. Samotná obsluha volby položek je implementována již existující metodou, která reaguje na výběr možnosti a vrací její název. Dle názvu se pak rozhodne, které proměnné je třeba změnit a o případné změně položek v nabídce.

Základ této aktivity obsahuje také uchovávání potřebných parametrů pro správné pokračování aplikace po rotaci zařízení. Při ní je celá aktivita automaticky ukončena a opětovně spuštěna. Takto by se aplikace po rotaci zařízení dostala opět do stavu prvního spuštění aktivity. Z toho důvodu je doimplementována metoda, jež se standardně spouští při ukončování aktivity. Ukládají se proměnné rozhodující o aktuálním typu zobrazení (2D a 3D), o rotaci a přiblížení molekuly ve 3D vizualizaci. Vždy po spuštění aktivity se kontroluje, jestli nejsou tyto proměnné nastaveny, což by znamenalo, že došlo k rotaci a je třeba vše vykreslit s jejich použitím. Tím se aktivita po rotaci zařízení dostane do stejného stavu jako v předchozí poloze.

7.6 2D vizualizace

Po vytvoření třídy zajišťující 2D vykreslování se volá metoda, jež primárně slouží k získání poměru zapsané molekuly k molekule vykreslované, která se celá vejde na obrazovku. Pro jeho získání se prochází celé pole atomů a porovnáváním jeho x a y souřadnic se získají jejich minimální a maximální hodnoty. Jejich vzájemným odečtením je zjištěna výška a šířka molekuly (rozsah hodnot). Těmito rozsahy je vydělena šířka a výška displeje, od nichž je

odečtena okrajová část. Zvolí se nižší hodnota a ta se dále používá jako násobitel pro všechny původní hodnoty molekuly. Kromě poměru je také vypočítáváno posunutí molekuly. Základní hodnota posunutí je pro obě osy rovna minimální hodnotě násobené číslem -1. U osy, pro kterou vyšel nižší násobitel, se k této hodnotě pouze přičte velikost volného místa od okraje displeje, což je 20 bodů. Pro druhou osu je k základnímu posunutí přičtena ještě hodnota odpovídající polovině rozdílu velikosti vykreslovací plochy a velikosti molekuly na dané ose.

Ve výše popsané metodě se prochází celé pole atomů kvůli získání jejich souřadnic. Zároveň s tím také dochází k určování barvy atomů dle jejich chemické značky. Definice barev pro jednotlivé prvky periodické tabulky je rozložena do dvou jednotlivých polí. V prvním jsou definovány všechny prvky a ve druhém jednotlivé barvy ve formátu RGB, přičemž každá složka je zapsána ve vlastním sloupci. Po načtení prvku aktuálního atomu se vyhledá totožný zápis v tabulce všech prvků. Jeho index se uloží do samotného Atomu. Konkrétní hodnoty barev se při vykreslení načtou z tabulky právě dle přiřazeného indexu. Pokud se nejedná o prvek periodické tabulky, ale o nějakou sloučeninu, tak je vybrána poslední barva z celé tabulky.

K vykreslování se využívá třídy Canvas. První se prochází celé pole vazeb. Je jim jednotně nastavena bílá barva. Vykreslují se pomocí čar, přičemž jejich počáteční a koncový bod je dán právě souřadnicemi vazby. Ty jsou ještě přenásobeny poměrem a přičteny k nim hodnoty posunutí. Kromě souřadnic se také ověřuje, o jakou vazbu se jedná. Pokud se jedná o dvojnou, je vykreslena pomocí dvou rovnoběžných čar. Před jejich vykreslením je použitím funkce tangents vypočten úhel, který vazba svírá s osou x. Hodnotami pro tuto funkci jsou rozdíly hodnot počátečních a koncových bodů na osách x a y. Přičtením 90° je získán úhel k němu kolmý. Předem definovaná vzdálenost obou vazeb je hodnotou přepony nově vzniklého pravoúhlého trojúhelníku. Na jejím základě jsou funkcemi sinus a cosinus vypočteny hodnoty protilehlé a přilehlé stěny trojúhelníku. Přičtením těchto hodnot k původním souřadnicím vazby vznikne první vazba. Odečtením je pak získána i druhá. Pokud by byl proveden posun jen pro jednu vazbu a druhá zůstala na původních souřadnicích, nevycházela by pak celá dvojná vazba ze středu atomu, ale byla by posunutá v závislosti na jejím úhlu.

Po vykreslení všech vazeb obsažených v seznamu se začnou procházet jednotlivé atomy. Nejprve je vždy nastavena černá barva, kterou je na pozici atomu vykreslen kruh. Tím se uvolní z místa, kde bude zapsána chemická značka prvku, všechny procházející vazby. Následně je z pole barev vybrána ta, jež se nachází na pozici, která je uložena v proměnné barva přímo v daném atomu. Doprostřed volného kruhu se zvolenou barvou zapíše chemická značka atomu.

Kromě samotné molekuly se na horní část displeje vykreslí ještě posuvník umožňující přibližování a oddalování kamery. Pro jeho funkci je definován Listener reagující na případné doteky uživatele na displej. Pokud je detekován dotyk v horní části displeje, kde se nachází posuvník, jsou zjištěny přesné souřadnice dotyku. Dle pozice na ose x je vypočítána hodnota přiblížení a hodnota posuvníku. Molekulu je možno zvětšit na pětinašobek původní hodnoty. To ji však dostane mimo vykreslovací oblast. K posunutí zobrazené plochy slouží proměnné, které

se přičítají k souřadnicím všech vykreslovaných prvků kromě posuvníku. Maximální hodnoty těchto proměnných jsou omezeny na velikost displeje vynásobenou rozdílem poloviny přiblížení. Při posunu o tyto hodnoty do všech stran se bude vždy nacházet zobrazovaná oblast v rozmezí velikosti displeje násobeného přiblížením. Pro dodržení tohoto pravidla se při oddalování molekuly vždy hodnoty určující posunutí korigují. Přiblížení molekuly je možné i pohybem dvou prstů k sobě či od sebe.

7.7 3D vizualizace

Podobně jako u 2D zobrazení i zde se nejprve provádí přepočítání poměru. Kvůli jinému systému souřadnic nelze použít dřívější přepočítání. Opět je tedy procházeno celé pole atomů a zjišťuje se rozměr molekuly, tentokrát však dle velikostí všech tří hodnot (x,y,z) . Výslednými rozsahy se vydělí číslo 2, které odpovídá velikosti koordinátového systému u OpenGL. Potřebným poměrem je nejmenší výsledná hodnota. Naprosto stejným způsobem se dopočte i posunutí na jednotlivých osách. Pro urychlení vykreslování jsou souřadnice všech atomů již roznásobeny, posunuty a vloženy do samostatného pole. Jelikož se pro 3D vizualizaci využívá totožného seznamu jako při 2D, která se vždy po zvolení vstupního souboru spustí jako první, není zapotřebí již znovu zjišťovat a doplňovat barvy jednotlivých atomů. Odlišnou částí však je zjišťování minimální vzájemné vzdálenosti jednotlivých atomů, které slouží k následnému určení jejich velikosti. Při tomto procesu je nutné porovnat vzájemně všechny atomy, což při značně velkých molekulách může na reálném zařízení trvat i v řádech sekund. Z toho důvodu se toto porovnávání využívá jen u molekul obsahujících maximálně 1000 atomů. Jejich nejmenší vzdálenost je zmenšena na pětinu a uložena jako základ pro výpočet poloměru atomů. V případě větších molekul je základ roven exponenciální funkci velikosti atomů s exponentem 0,5.

Před vykreslováním se vypočtou jednotlivé body koule o poloměru 1, jež bude reprezentovat při vizualizaci atomy. Jednotlivé body budou sloužit pro vytváření polygonů, jimiž bude znázorněn povrch koule. Všechny vypočtené body se vloží do bufferu. Do něj se kromě bodů jedné koule pro všechny atomy vloží i přepočtené (vynásobené poměrem a posunuté) souřadnice všech vazeb. Dvojnásobné vazby jsou opět rozděleny na dvě jednotlivé a posunuty navzájem na ose x a y . Vazby budou kresleny pomocí čar, což urychlí celé vykreslování. Pokud by pro jejich reprezentaci bylo využito válců, zvýšily by se tím značně výkonové požadavky na používané zařízení a snížila by maximální velikost plynule vykreslovaných molekul na stejném zařízení.

Při samotném vykreslování se nastaví pozice kamery a světla, jejich pozice bude fixní a po celou dobu vizualizace nezávisle na rotaci molekuly stále stejná. Jedinou výjimku tvoří pohyb kamery na ose z , což umožní přibližování objektu. Pozice kamery je definována v bodě $[0,0,5]$ a při maximálním přiblížení $[0,0,0.1]$. Světlo je umístěno vůči kameře napravo nahoru. Po umístění těchto dvou prvků je uložena hodnota transformační matice (poloha souřadného systému) a celý vykreslovaný systém je pootočen o případnou hodnotu rotace dle osy x a y .

Původní hodnota transformační matice se opět načítá až po vykreslení všech prvků na jejich pozice ve scéně.

Vykreslování se dále větví dle toho, jaký je aktuálně zvolen zobrazovací model molekuly. Tato informace je uložena ve vlastní proměnné a na základě její hodnoty se dále pokračuje v umisťování jednotlivých prvků do scény. Základním modelem je kuličkový, jenž znázorňuje jak vazby tak i samotné atomy. Kalotový model znázorňuje atomy, ovšem s větším poloměrem, kdy vazby jsou nahrazeny vzájemným průnikem atomů. Posledním modelem je drátěný, kde se vykreslí pouze vazby.

Při umisťování atomů do scény je vždy znovu procházen celý jejich seznam, z něž se získávají údaje o barvě a počtu jeho vazeb. Velikost atomu se určuje na základě již vypočtené základní velikosti, k níž se ještě přičítá 20% této velikosti za každou jeho vazbu. Před samotným jeho kreslením se uloží hodnota transformační matice, posune celý souřadný systém na pozici již dříve připravenou do speciálního pole a provede se také změna velikosti systému. Po nastavení pozice v bufferu na umístění předchystané koule a jejím vykreslení dostaneme atom o zvoleném poloměru na specifikované pozici. Pro správný výpočet osvětlení je nutné definovat i normálové vektory. Ty však odpovídají hodnotám jednotkové koule použité jako základ pro atom, takže není zapotřebí je zvlášť dopočítávat. Před vykreslením dalšího atomu je vždy nutné načíst poslední uloženou hodnotu transformační matice. Tím se opět dostane souřadný systém do pozice po rotaci celého objektu. U drátěného modelu se vykreslování atomů vynechává, zatímco u kalotového se pouze poloměr atomu násobí třemi.

Před umisťováním vazeb do scény je vypnuto osvětlení, protože jsou znázorněny pouze čarami s nastavenou šířkou. To by způsobovalo problémy u normálových vektorů a výsledný osvětlovací efekt by byl jiný než u atomů. Opět je procházeno celé pole vazeb, ale získává se z něj pouze informace o tom, zda se jedná o dvojnou či jednoduchou vazbu a o barvách obou atomů, jež spojuje. Souřadnice vazby jsou totiž již uloženy v bufferu, který stačí vždy pouze nastavit na patřičnou pozici. Každá vazba je rozdělena na poloviny. Obě jsou vykresleny zvlášť a obarveny dle atomu, z něž vycházejí. Po vykreslení všech vazeb je znovu zapnuto osvětlení scény. Vazby jsou vykreslovány pro drátěný a kuličkový model.

Celým modelem lze tažením prstu po displeji natáčet kolem dvou os. K tomu je definován listener reagující na dotek displeje. Ten si uchovává vždy hodnotu souřadnic posledního dotyku a porovnává je s aktuálními. Porovnáním těchto parametrů je získán směr pohybu a dle něj se mění úhel natočení na osách x a y . Kromě manuálního ovládání rotace lze také v menu zapnout automatickou rotaci. V takovém případě je postupně zvyšován úhel otočení kolem osy y a model automaticky rotuje. Při jejím zapnutí je však zapotřebí změnit režim překreslování, který udává, kdy se celá scéna opětovně vykresluje. V tomto případě je nutné ho nastavit na plynulé vykreslování. Při vypnutí automatické rotace se nastaví režim na spouštění opětovného vykreslování pouze při dotyku displeje či stisku kláves.

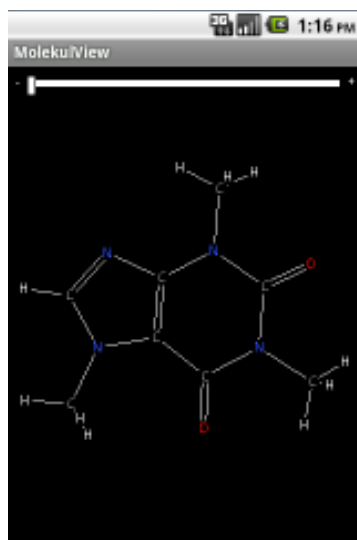
8 Testování aplikace

Jelikož k testování aplikace bylo k dispozici pouze jediné reálné testovací zařízení, byly některé testy prováděny pouze v emulátoru. Ten umožňuje vlastní volbu technických parametrů zařízení, z nichž hlavní je možnost nastavení rozlišení displeje. Testované parametry z emulátoru budou dále srovnány s hodnotami z reálného zařízení. Po testu na reálném zařízení bude aplikace také porovnána s aplikací Jmol spuštěným na totožném zařízení.

8.1 Testování v emulátoru

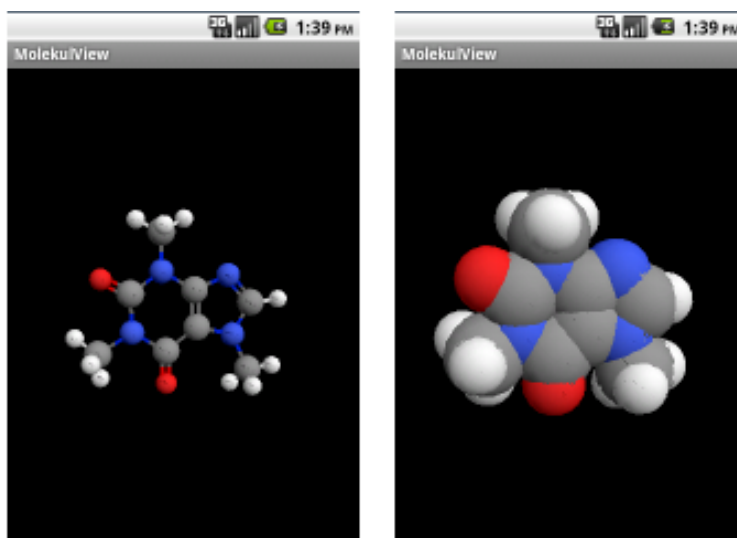
Kromě otestování základního chování aplikace bylo v emulátoru testováno hlavně zobrazování aplikace na displeji s různými rozlišeními.

Minimální požadovaná verze systému Android pro funkci všech implementovaných vlastností je Android 2.2. V nižších verzích není možno využít použitý systém pro multitouch. Také se v této verzi změnila adresa paměťové karty, takže na zařízení s nižší verzí by aplikace nenačetla její adresářovou strukturu.



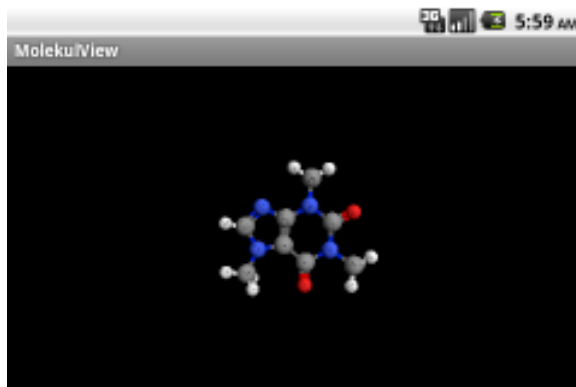
Obrázek 8.1: 2D vizualizace kofeinu na prvním simulovaném zařízení

První simulované zařízení má operační systém Android ve verzi 2.2. Velikost displeje byla nastavena na 320x480 bodů s jemností 181 bodů na palec, což odpovídá hodnotám reálného zařízení HTC Hero. Dle něj byla také nastavena velikost operační paměti na 288MB.



Obrázek 8.2: 3D vizualizace kofeinu na prvním simulovaném zařízení

Během simulace prvního testovacího zařízení byla aplikace plně funkční, včetně správného vykreslování molekuly, což zahrnuje správné výpočty poměru a posunutí pro danou velikost displeje.



Obrázek 8.3: 3D vizualizace kofeinu na prvním simulovaném zařízení po rotaci zařízení

Další simulovaná zařízení měla hodnoty operační paměti nastaveny na 512MB a 1024MB a různá nastavení velikosti displeje. Konkrétní další testované velikosti displeje byly 854x480px a 1280x720px. Vizualizovaná molekula se zobrazovala vždy shodně v poměru dle velikosti displeje.

Na chování aplikace v emulátoru nemělo nastavení operační paměti žádný vliv. Nezávisle na její velikosti nebyla rotace molekul plynulá. Pro menší molekuly (např. molekula kofeinu viz. obrázek 8.1 až 8.3) byla odezva při 3D zobrazování do 2s. U 2D zobrazení byla reakce okamžitá. Doba potřebná pro první zobrazení molekuly, což znamená součet času pro samotné vykreslení a přípravu všech potřebných výpočtů (např. poměr a posunutí), byla shodná s tou, kterou zařízení vykazovalo při pouhém následném vykreslování. Při vizualizaci větší molekuly (celkem 5247 atomů) byla doba prvního zobrazení ve 2D rovna 12s. Z čehož 10s trvaly počáteční výpočty a 2s samotné vykreslení. 3D vizualizace trvala znatelně déle, potřebná doba dosahovala až 60s, přičemž doba pro výpočet byla shodná jako u 2D.

8.2 Testování na reálném zařízení



Model: Sony Ericsson Xperia Ray (ST18i)

Processor: 1GHz Snapdragon

Grafický čip: Adreno 205

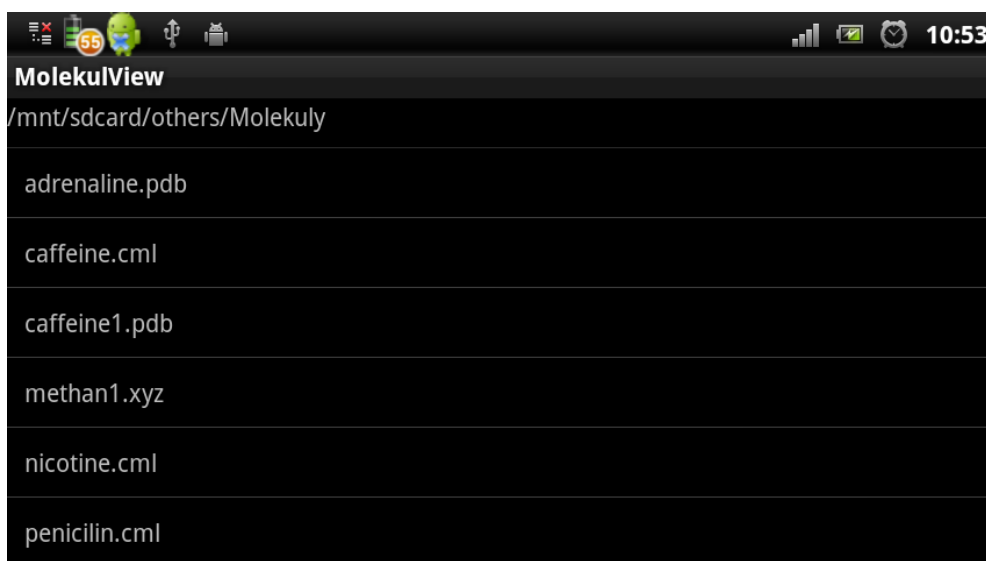
Operační paměť: 512MB

Operační systém: Android 2.3.4 Gingerbread

Rozlišení displeje: 854x480 pixelů

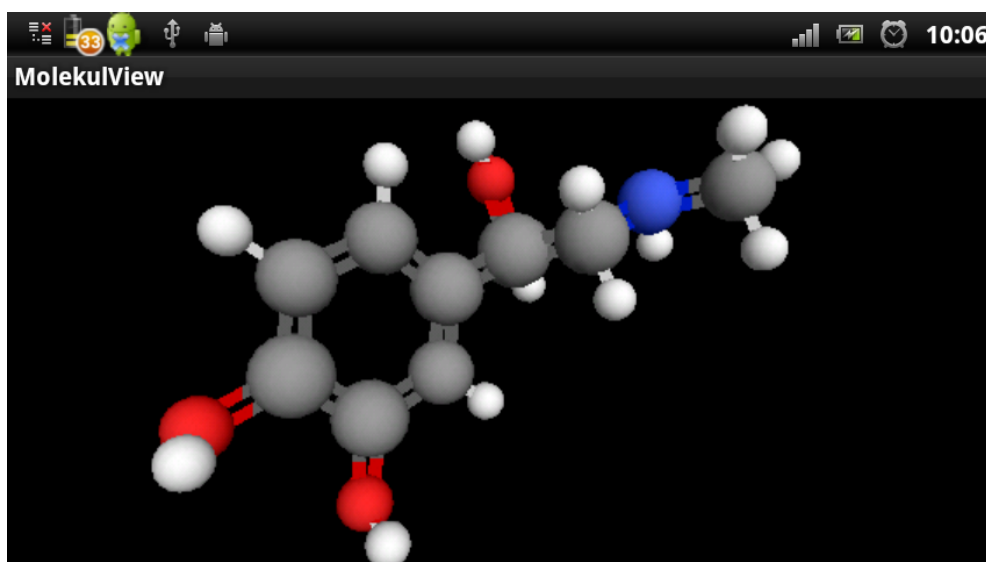
Obrázek 8.4: Testovací zařízení a jeho technické parametry

Pro testování bylo k dispozici jedno reálné zařízení. Jednalo se o model Xperia Ray od firmy Sony Ericsson. Jeho technické parametry jsou uvedeny v obrázku 8.4. Oproti emulátoru, jenž nepodporuje multitouch, bylo možno již provést i test přibližování molekuly pohybem dvou prstů. Stejně jako ostatní i tato implementovaná vlastnost je funkční.



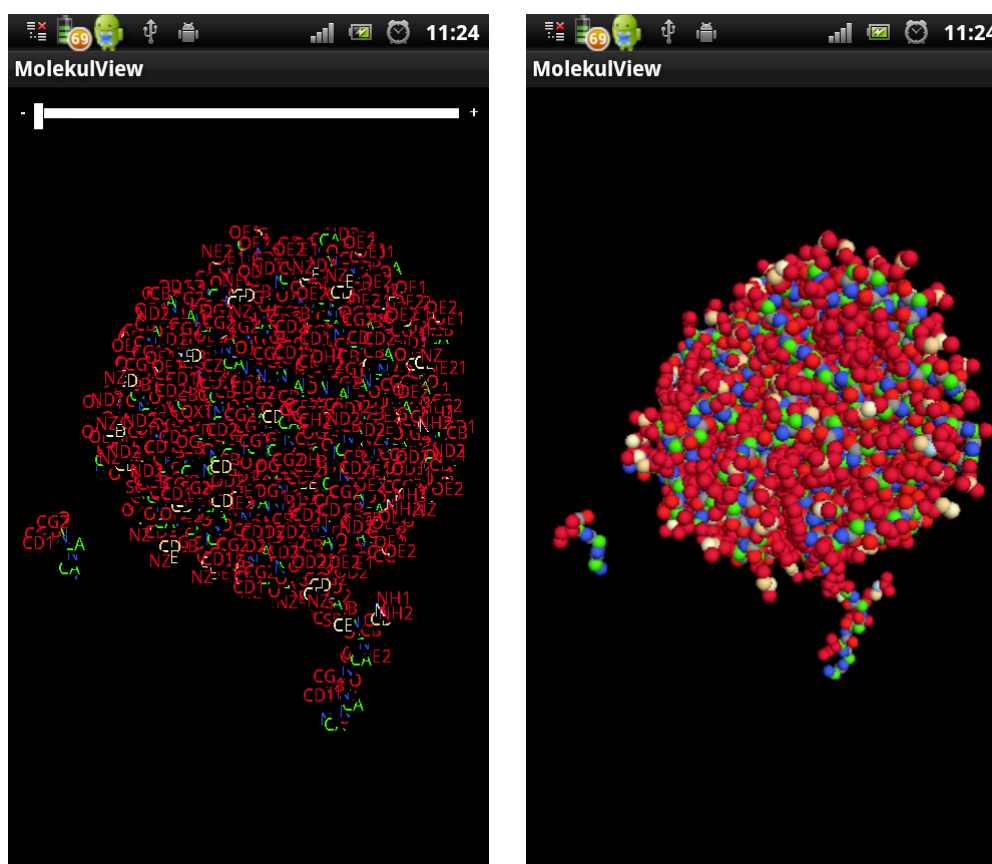
Obrázek 8.5: Souborový prohlížeč aplikace

V zařízení byla správně načtena adresářová struktura přeměťové karty a nebylo možno se dostat mimo ní, což bylo v souladu s nedostatečným oprávněním aplikace. V případě, že by tomu nebylo zabráněno došlo by k chybě a neočekávanému ukončení aplikace. Průchod celou adresářovou strukturou i volba podporovaného souboru (jiný není možno otevřít) fungoval bez neočekávaného chování či ukončení aplikace.



Obrázek 8.6: Vizualizace molekuly adrenalinu – kuličkový model

Při vizualizaci molekul se vždy v základním zobrazení vykreslila celá molekula, což znamená, že byl správně vypočten poměr její velikosti oproti původnímu souboru a také posunutí. Během vizualizaci molekuly kofeinu(totožný vstupní soubor jako v emulátoru) i jakékoli jiné s podobným počtem atomů reagovala aplikace ve všech zobrazeních okamžitě bez jakéhokoli zpoždění a to i ve 3D režimu se zapnutou automatickou rotací. Při vykreslení větší molekuly(celkem 5247 atomů, stejný vstupní soubor jako v emulátoru) výpočty pro její umístění již trvaly 2s. Samotná 2D vizualizace(přibližování a posun) pak již byla plynulá. Příprava 3D vizualizace zabrala méně než 2s. Ovládání molekuly po vykreslení však reagovalo s maximální odezvou do 1s a automatická rotace nebyla plynulá



Obrázek 8.7: 2D a 3D(kalotový model) vizualizace molekuly selenofosfátu

8.3 Srovnání testů

Aplikace byla funkční jak v emulátoru, tak i na reálném zařízení, ale v emulátoru nebylo možno ověřit funkčnost přibližování molekuly metodou využívající multitouch. Zatímco emulátor vykazoval již u menších molekul, které obsahovaly řádově desítky atomů, značné problémy s plynulým během aplikace, na reálném zařízení se projevovalo zpoždění při vykreslování od neměřitelných hodnot do 1s v závislosti na velikosti molekuly.

Podstatně nižší schopnost plynulého vykreslování u emulátoru je způsobena pouhou virtualizací operačního systému Android a také absencí hardwarové akcelerace grafiky, které je zastoupena na reálných zařízeních. Časové rozdíly způsobené rozdílnou konfigurací (reálné a virtuální zařízení) byly dojnásobné až pětinasobné dle velikosti molekuly, což plyne z doby potřebné pro výpočty před samotným vykreslováním molekuly. Rozdíl mezi dobou potřebnou pro vykreslení celé molekuly byl již podstatně větší.

Tabulka 8.1: Potřebná doba pro vizualizaci testovaných molekul

	Molekula kofeinu - 24 atomů				Molekula selenofosfátu - 5247 atomů			
	Příprava		Vykreslování		Příprava		Vykreslování	
	2D	3D	2D	3D	2D	3D	2D	3D
Emulátor	<1s	<1s	<1s	2s	10s	10s	2s	60s
Reálné zařízení	<1s	<1s	<1s	<1s	2s	1s	<1s	1s

8.4 Porovnání aplikace s Jmol

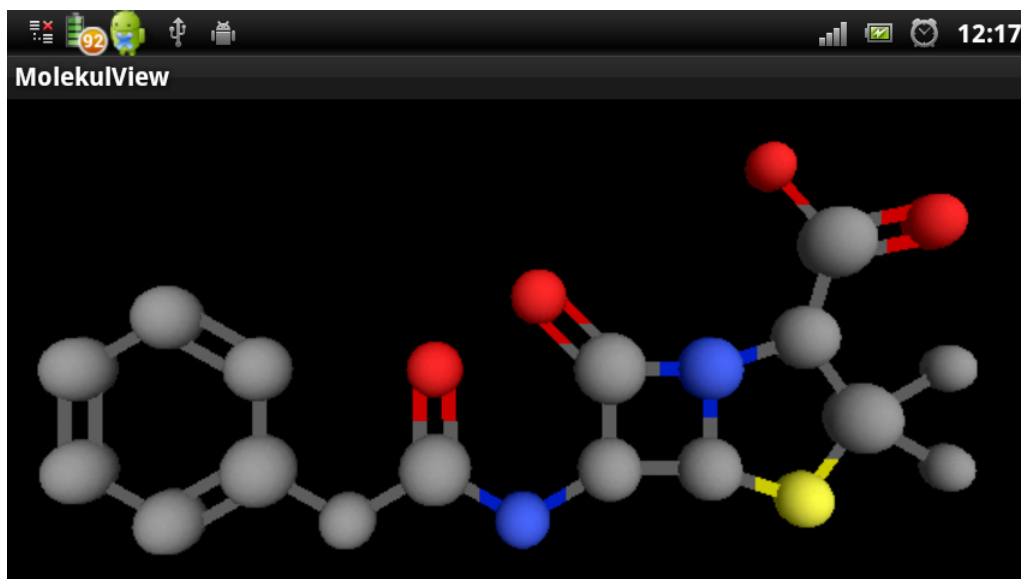
Ke srovnání vytvořené aplikace byl vybrán program Jmol a jeho verze dostupná pro platformu Android. Ta sice obsahuje veškeré vlastnosti programu Jmol dostupného pro jakoukoli platformu, ale přístup k většině z nich je pouze pomocí textových příkazů, takže pro neznalého uživatele zůstávají bez dalšího studia dokumentace či příručky nedostupné. Obě aplikace shodně vyžadují minimální verzi operačního systému Android 2.2.

Hlavní výhodou aplikace Jmol je databáze různých chemických sloučenin, díky nimž dokáže doplnit vazby i molekulám, jejichž vstupní soubor je nedefinuje. Kromě toho má přímý přístup do databáze PDB a NIH/NCI z nějž může stahovat molekuly. Všechny tři formáty aktuálně implementované ve vytvořené aplikaci jsou zahrnuty i v Jmolu.

Zatímco aplikace umožňuje snadné zvětšování molekuly, či zapnutí automatické rotace, u Jmolu lze toto v aktuální verzi provést pouze pomocí textového příkazu. Přepínání

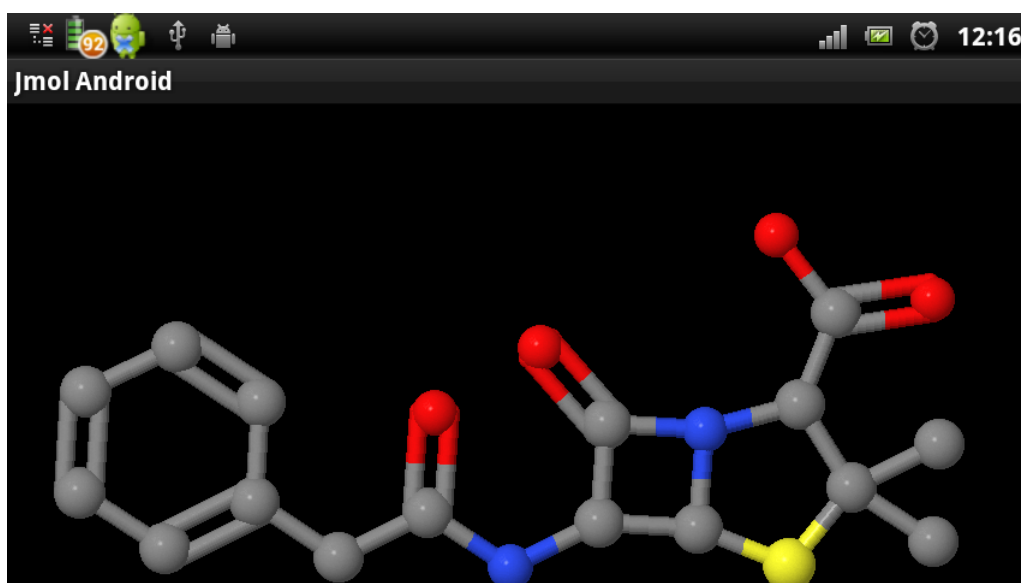
zobrazovacích modelů je shodně umístěno přímo v menu. Jmol neobsahuje 2D zobrazení, ale má více modelů pro 3D zobrazení molekuly.

Grafické výstupy při použití stejných modelů jsou značně podobné (viz obrázky 8.8. a 8.9). Neliší se ani volba barev, protože obě aplikace využívají stejného barevného schématu. Rozdílné je akorát schéma osvětlení a umístění molekuly v rámci zobrazovací plochy. U Jmolu se často stává, že zobrazovaná molekula je na jedné straně displeje již mimo zobrazovanou plochu, zatímco na druhé je stále ještě volné místo. Implementovaná aplikace se automaticky přizpůsobuje poloze zařízení, ale Jmol je implementován pouze pro jedinou polohu a aplikace neprovede otočení dokonce ani tehdy, když se celé zařízení otočí o 180° kolem horizontální osy.



Obrázek 8.8: 3D vizualizace molekuly penicilinu v implementované aplikaci

Samotné zpracovávání vstupního souboru bylo na testovacím zařízení v obou aplikacích stejně rychlé, ale během 3D zobrazení molekuly selenofosfátu byla doba odezvy u Jmolu poloviční. Jmol je díky tomu vhodnější hlavně k vizualizaci větších molekul a také k zobrazování molekul, jejichž vstupní soubor není kompletní. Tedy soubor, který obsahuje pouze zápis jednotlivých atomů.



Obrázek 8.9: 3D vizualizace molekuly penicilinu v aplikaci Jmol

9 Závěr

V průběhu této práce vznikla aplikace schopná vizualizace molekul, jejichž geometrie je zapsána ve třech různých vstupních formátech. Implementační platformou je Android, přičemž minimální požadovaná verze pro běh aplikace na zařízení je 2.2. Při testování v emulátoru se aplikace chovala značně pomalu i při vykreslování malých molekul, ale během testů na reálném zařízení již běžela pro molekuly obsahující do 2000 atomů plynule. Aktuálně podporovanými vstupními formáty jsou PDB, CML a XYZ.

V porovnání s nejznámějším zástupcem již existujících aplikací na tuto platformu, konkrétně se jednalo o Jmol, dosahovala při vykreslování molekul srovnatelných výsledků. Její hlavní předností oproti této aplikaci je jednoduchost ovládání. Jmol ve svém základu obsahuje mnoho dalších funkcí včetně různých měření molekul, ale ty jsou dostupné pouze přes textové příkazy, jež musí uživatel znát nebo dohledat v dokumentaci. Aplikace implementovaná v průběhu této diplomové práce obsahuje všechny ovládací prvky přímo na displeji a veškeré činnosti se provádí na základě grafických prvků (přibližování pomocí posuvníku, položky v menu). Jmol je však rychlejší pro molekuly obsahující nad 2000 atomů a také obsahuje databázi sloučenin na jejímž základě dokáže vykreslit i vazby, jež nejsou ve vstupním souboru přímo definovány.

Implementovaná aplikace lze dále rozšiřovat o nové formáty vstupních souborů. Další možností rozšíření je implementace databáze vazeb podobná té z Jmolu, nebo další shodná vlastnost, kterou je načítání souborů přímo z některé z internetových databází molekul. Také by bylo možno přidat další vykreslovací 3D model.

10 Použitá literatura

- [1] Atomic Coordinate Entry Format Version 3.3 [online]. 2011 [cit. 2012-04-30]. Dostupné z: <http://www.wwpdb.org/documentation/format33/v3.3.html>
- [2] Chemical Markup Language - Molecular Convention [online]. 2011 [cit. 2012-04-30]. Dostupné z: <http://www.xml-cml.org/convention/molecular>
- [3] PREADMET. MDL mol and sd file [online]. 2010 [cit. 2012-04-30]. Dostupné z: http://preadmet.bmdrc.org/index.php?option=com_content&view=article&id=67&Itemid=84
- [4] Open Babel: The Open Source Chemistry Toolbox [online]. 2008 [cit. 2012-04-30]. Dostupné z: <http://openbabel.org>
- [6] RasMol and OpenRasMol [online]. 2000 [cit. 2012-04-30]. Dostupné z: <http://www.rasmol.org/>
- [7] MOLEGRO. Molegro Molecular Viewer [online]. 2005 [cit. 2012-04-30]. Dostupné z: <http://www.molegro.com/mmv-product.php>
- [8] Jmol: an open-source Java viewer for chemical structures in 3D [online]. 2000 [cit. 2012-04-30]. Dostupné z: <http://jmol.sourceforge.net/>

11 Přílohy

Seznam příloh

Příloha A: Uživatelská příručka.....	II
Příloha B: Adresářová struktura přiloženého DVD.....	III

Uživatelská příručka

Aplikace slouží k vizualizaci molekul na základně vstupních souborů popisujících jejich geometrii. Vstupní soubor musí být zapsán pomocí znakové sady UTF-8. Podporované vstupní formáty jsou PDB, CML a XYZ. Soubory s jinými příponami aplikace neotevře. Zařízení, na němž je spuštěna, musí obsahovat operační systém Android minimálně ve verzi 2.2.

První obrazovkou po spuštění je jednoduchý souborový prohlížeč, jenž automaticky načte paměťovou kartu. Pomocí něj lze procházet adresářovou strukturou paměťové karty, dokud není zvolen požadovaný soubor. Vizualizace není zahájena, dokud není vybrán vstupní soubor ve formátu specifikovaném výše.

Vizualizace molekuly nejprve probíhá ve 2D prostoru. Molekulu je možno přibližovat buď posuvníkem ve vrchní části obrazovky, nebo pomocí přibližování a oddalování dvou prstů. Při přiblížení molekuly lze měnit zobrazovanou část tažením prstu po displeji zařízení. Na 3D vizualizaci lze přepnout pomocí tlačítka pro zobrazení menu, kde se vybere položka 3D.

Během 3D vizualizace je možno molekulou tažením prstou otáčet dle její osy X a Y. Přibližování je realizováno opět pomocí dvou prstů nebo vodorovným tahem ve vrchní části zobrazovací plochy. Pomocí tlačítka menu lze vyvolat nabídku možností pro 3D vizualizaci. Ta obsahuje přepnutí modelu molekuly (kuličkový, kalotový a drátěný), znutí automatické rotace dle osy Y a možnost přechodu na 2D vizualizaci.

Kdykoli během vizualizace se lze dostat zpět na výběr vstupního souboru pomocí tlačítka zpět.

Příloha A: Uživatelská příručka

/Aplikace	Obsahuje spustitelnou aplikaci a zdrojové kódy
/Diplomová práce	Obsahuje diplomovou práci a její zadání
/Dokumentace	Programátorská příručka
/Vstupní soubory	Adresář obsahuje vstupní soubory molekul

Příloha B: Adresářová struktura přiloženého DVD